

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de: INGENIERO DE  
SISTEMAS**

**TEMA:**

**ANÁLISIS, DISEÑO, DESARROLLO E INTEGRACIÓN DEL MÓDULO DE  
ASIGNACIÓN DE AULAS Y ESPACIOS FÍSICOS PARA EL SISTEMA UPS  
SCHEDULE PARA LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE  
QUITO.**

**AUTOR:**

**IVÁN MAURICIO ROMERO MEJÍA**

**DIRECTOR:**

**RODRIGO EFRAÍN TUFIÑO CÁRDENAS**

**Quito, Mayo 2015**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE TITULACIÓN**

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

Quito, Mayo del 2015.

---

Iván Mauricio Romero Mejía  
1717773616

## **DEDICATORIA**

A mi Madre, por darme la vida y todo lo demás.

Iván Mauricio Romero Mejía

## ÍNDICE

<b>INTRODUCCION.....</b>	<b>1</b>
<b>CAPÍTULO 1 .....</b>	<b>2</b>
<b>1. MARCO REFERENCIAL Y TEÓRICO.....</b>	<b>2</b>
<b>1.1. Planteamiento del problema.....</b>	<b>2</b>
<b>1.2. Objetivos.....</b>	<b>5</b>
1.2.1. Objetivo general.....	5
1.2.2. Objetivos específicos.....	5
<b>1.3. Justificación .....</b>	<b>6</b>
<b>1.4. Alcance.....</b>	<b>6</b>
<b>1.5. Marco Teórico .....</b>	<b>7</b>
1.5.1. Programación orientada a objetos.....	7
1.5.2. Java .....	11
1.5.3. Metodología SCRUM.....	12
<b>CAPÍTULO 2 .....</b>	<b>15</b>
<b>2. ANÁLISIS Y DISEÑO .....</b>	<b>15</b>
<b>2.1. Análisis de viabilidad.....</b>	<b>15</b>
2.1.1. Viabilidad técnica .....	15
2.1.2. Viabilidad operativa.....	16
2.1.3. Viabilidad económica .....	16
<b>2.2. Análisis de requerimientos .....</b>	<b>17</b>
2.2.1. Definiciones y abreviaturas.....	17
2.2.2. Descripción general .....	18
2.2.3. Perspectiva del producto .....	18
2.2.4. Funciones del producto.....	18
2.2.5. Características de usuarios .....	19
2.2.6. Suposiciones y dependencias .....	20
2.2.7. Requisitos futuros .....	20
2.2.8. Requisitos específicos.....	20
2.2.8.1. Acceder al sistema .....	20
2.2.8.2. Modificar clave de acceso.....	20

2.2.8.3. Visualizar estadísticas de asignación .....	21
2.2.8.4. Gestión materias sin aula .....	21
2.2.8.5. Eliminar asignación de aula .....	21
2.2.8.6. Generación de reporte por docente.....	21
2.2.8.7. Generación de reporte por grupo .....	22
2.2.8.8. Generación de reporte por aula .....	22
2.2.9. Atributos del sistema .....	22
<b>2.3. Diseño .....</b>	<b>22</b>
2.3.1. Diagrama de casos de uso .....	23
2.3.2. Diagrama de componentes .....	27
2.3.3. Diagrama de bloques .....	28
2.3.4. Prototipos de interfaces .....	29
2.3.4.1. Ingreso al sistema .....	29
2.3.4.2. Cambio de contraseña .....	29
2.3.4.3. Inicio.....	30
2.3.4.4. Espacios físicos.....	31
2.3.4.5. Materias sin aula .....	32
2.3.4.6. Reporte docente .....	33
2.3.4.7. Reporte aula .....	33
2.3.4.8. Reporte grupo .....	34
<b>CAPÍTULO 3 .....</b>	<b>35</b>
<b>3. CONSTRUCCIÓN .....</b>	<b>35</b>
<b>3.1. Plataformas y herramientas .....</b>	<b>35</b>
3.1.1. Netbeans IDE .....	35
3.1.2. JDK .....	36
3.1.3. Modelo Vista Controlador (MVC) .....	36
3.1.4. PostgreSQL .....	36
3.1.5. Algoritmo MD5 .....	37
3.1.6. ITextPdf.....	37
3.1.7. JCommon.....	37
3.1.8. JFreeChart .....	38
3.1.9. Power Architec.....	38
3.1.10. Pencil .....	38
3.1.11. DBVisualizer.....	38

3.1.12. Visual Paradigm para UML .....	39
3.1.13. Notepad++ .....	39
3.1.14. Dia.....	39
<b>3.2. Diagrama conceptual de la base de datos UPSSchedule .....</b>	<b>40</b>
3.2.1. Diccionario de datos .....	41
3.2.1.1. Tbl_distributivo .....	41
3.2.1.2. Tbl_carrera .....	41
3.2.1.3. Tbl_dia .....	42
3.2.1.4. Tbl_estructura .....	42
3.2.1.5. Tbl_estado_horario.....	43
3.2.1.6. Tbl_grupo.....	43
3.2.1.7. Tbl_hora .....	43
3.2.1.8. Tbl_malla .....	43
3.2.1.9. Tbl_nivel .....	44
3.2.1.10. Tbl_horario .....	44
3.2.1.11. Tbl_mencion .....	45
3.2.1.12. Tbl_materia.....	45
3.2.1.13. Tbl_rango_aulas .....	46
3.2.1.14. Tbl_periodo.....	46
3.2.1.15. Tbl_tipo_estructura .....	46
3.2.1.16. Tbl_tipo_aula.....	46
3.2.1.17. Tbl_usuario .....	47
<b>3.3. Estructura de paquetes y clases .....</b>	<b>47</b>
3.3.1. Diagrama de paquetes.....	48
3.3.2. Paquete Vista.....	49
3.3.2.1. Clase Login .....	50
3.3.2.2. Clase Principal .....	51
3.3.2.3. Clase CambiarContrasena .....	53
3.3.2.4. Clase RangoAulas .....	53
3.3.2.5. Clase RepAula .....	54
3.3.2.6. Clase RepDocente .....	54
3.3.2.7. Clase RepGrupo.....	55
3.3.3. Paquete Controlador .....	56
3.3.3.1. Clase ConLogin .....	57

3.3.3.2. Clase ConChartCampus .....	57
3.3.3.3. Clase ConChartCarrera .....	57
3.3.3.4. Clase ConChartBloques .....	58
3.3.3.5. Clase ConRangoAulas .....	58
3.3.3.6. Clase ConEspaciosFisicos .....	59
3.3.3.7. Clase ConGestion .....	60
3.3.3.8. Clase ConReporteAula.....	60
3.3.3.9. Clase ConReporteDocente .....	61
3.3.3.10. Clase ConReporteGrupo .....	62
3.3.3.11. Clase ConGenerica .....	62
3.3.4. Paquete Modelo .....	63
3.3.4.1. Clase conexión.....	64
3.3.4.2. Clase modLogin.....	64
3.3.4.3. Clase modDia.....	64
3.3.4.4. Clase modHora .....	64
3.3.4.5. Clase modCarrera .....	64
3.3.4.6. Clase modMateria.....	64
3.3.4.7. Clase modAula .....	65
3.3.4.8. Clase SpringUtilities .....	65
3.3.4.9. Clase modEstadisticas .....	65
3.3.4.10. Clase modEspaciosFisicos .....	65
3.3.4.11. Clase modRangoAulas .....	65
3.3.4.12. Clase modReporteAula .....	66
3.3.4.13. Clase modReporteDocente .....	66
3.3.4.14. Clase modReporteGrupo.....	66
3.3.5. Paquete Persistencia.....	67
<b>3.4. Resultados .....</b>	<b>69</b>
3.4.1. Interfaz Login.....	69
3.4.2. Interfaz Cambiar Contraseña.....	69
3.4.3. Interfaz Inicio .....	70
3.4.4. Interfaz Materias sin aulas.....	71
3.4.5. Interfaz Espacios Físicos .....	72
3.4.6. Reporte por Aula.....	73
3.4.7. Reporte por Docente .....	74

3.4.8. Reporte por Grupo .....	75
<b>CAPÍTULO 4 .....</b>	<b>76</b>
<b>4. CARGA DE DATOS Y PRUEBAS .....</b>	<b>76</b>
<b>4.1. Carga de información a la base de datos UPSSchedule .....</b>	<b>76</b>
4.1.1. Tablas catálogo .....	76
4.1.2. Archivos UPS.....	79
4.1.3. Proceso de carga .....	80
4.1.3.1. Carga tbl_usuario.....	81
4.1.3.2. Carga tbl_estructura .....	82
4.1.3.3. Carga tbl_carrera .....	82
4.1.3.4. Carga tbl_materia .....	83
4.1.3.5. Carga tbl_Distributivo.....	84
4.1.3.6. Carga tbl_horario.....	85
<b>4.2. Pruebas .....</b>	<b>87</b>
4.2.1. Pruebas contra requerimientos .....	87
4.2.2. Pruebas de usabilidad .....	88
<b>CONCLUSIONES .....</b>	<b>93</b>
<b>RECOMENDACIONES .....</b>	<b>94</b>
<b>LISTA DE REFERENCIAS .....</b>	<b>95</b>
<b>ANEXOS.....</b>	<b>98</b>



## ÍNDICE DE FIGURAS

Figura 1. Módulos del proyecto UPS Schedule y responsables del desarrollo .....	4
Figura 2. Componentes de una clase.....	8
Figura 3. Ejemplo de herencia.....	10
Figura 4. Proceso de la Metodología Scrum.....	13
Figura 5. Diagrama de Casos de Uso. ....	23
Figura 6. Diagrama de Componentes programa Gestión de Espacios Físicos .....	27
Figura 7. Diagrama de Bloques programa Gestión de Espacios Físicos .....	28
Figura 8. Prototipo Interfaz Ingreso al Sistema.....	29
Figura 9. Prototipo Interfaz Cambio de contraseña.....	29
Figura 10. Prototipo Interfaz Inicio .....	30
Figura 11. Prototipo Interfaz Espacios Físicos .....	31
Figura 12. Prototipo Interfaz Materia sin Aula .....	32
Figura 13. Prototipo Interfaz Reporte Docente .....	33
Figura 14. Prototipo Interfaz Reporte Aula .....	33
Figura 15. Prototipo Interfaz Reporte Grupo.....	34
Figura 16. Diagrama Conceptual BDD UPSSchedule.....	40
Figura 17. Diagrama de Paquetes.....	48
Figura 18. Diagrama de Clases del Paquete Vista.....	49
Figura 19. Diagrama de clases del Paquete Controlador .....	56
Figura 20. Diagrama de Clases del Paquete Modelo .....	63
Figura 21. Diagrama Paquete Controlador .....	67
Figura 22. Interfaz Login – Sistema Gestión de Espacios Físicos .....	69
Figura 23. Interfaz Cambiar Contraseña – Sistema Gestión de Espacios Físicos.....	69
Figura 24. Interfaz Inicio – Sistema Gestión de Espacios Físicos .....	70
Figura 25. Interfaz Materias sin aulas – Sistema Gestión de Espacios Físicos .....	71
Figura 26. Interfaz Espacios Físicos – Sistema Gestión de Espacios Físicos .....	72
Figura 27. Reporte de aula – Sistema Gestión de Espacios Físicos.....	73
Figura 28. Reporte de docente – Sistema Gestión de Espacios Físicos.....	74
Figura 29. Reporte de grupo – Sistema Gestión de Espacios Físicos .....	75
Figura 30. Esquema carga tablas BDD UPSSchedule .....	80
Figura 31. Proceso de carga de tbl_usuario .....	81
Figura 32. Proceso de carga de tbl_carrera.....	82

Figura 33. Proceso de carga de tbl_materia .....	83
Figura 34. Proceso de carga de tbl_distributivo .....	84
Figura 35. Proceso de carga de la tbl_horario .....	85
Figura 36. Pregunta 1 – Prueba de Usabilidad .....	89
Figura 37. Pregunta 2 – Prueba de Usabilidad .....	89
Figura 38. Pregunta 3 – Prueba de Usabilidad .....	90
Figura 39. Pregunta 4 – Prueba de Usabilidad .....	90
Figura 40. Pregunta 5 – Prueba de Usabilidad .....	91
Figura 41. Pregunta 6 – Prueba de Usabilidad .....	91
Figura 42. Pregunta 7 – Prueba de Usabilidad .....	92
Figura 43. Pregunta 8 – Prueba de Usabilidad .....	92
Figura 44. Formato carga tabla tbl_usuario .....	98
Figura 45. Formato carga tabla tbl_estructura.....	98
Figura 46. Formato carga tabla tbl_carrera.....	99
Figura 47. Formato carga tabla tbl_materia.....	99
Figura 48. Formato carga tabla tbl_distributivo .....	100
Figura 49. Formato carga tabla tbl_horario .....	100
Figura 50. Reporte por aula .....	101
Figura 51. Reporte por docente .....	102
Figura 52. Reporte por grupo .....	103

## ÍNDICE DE TABLAS

Tabla 1. Costos de Implementación .....	16
Tabla 2. Requisito funcional 1 .....	20
Tabla 3. Requisito funcional 2.....	20
Tabla 4. Requisito funcional 3.....	21
Tabla 5. Requisito funcional 4.....	21
Tabla 6. Requisito funcional 5.....	21
Tabla 7. Requisito funcional 6.....	21
Tabla 8. Requisito funcional 7.....	22
Tabla 9. Requisito funcional 8.....	22
Tabla 10. Descripción caso de uso cambiar contraseña .....	23
Tabla 11. Descripción caso de uso visualizar estadísticas .....	24
Tabla 12. Descripción caso de uso asignar materias sin aula .....	24
Tabla 13. Descripción caso de uso gestionar espacios físicos.....	25
Tabla 14. Descripción caso de uso generar reporte.....	25
Tabla 15. Descripción de la tabla tbl_distributivo .....	41
Tabla 16. Descripción de la tabla tbl_carrera .....	41
Tabla 17. Descripción de la tabla tbl_dia.....	42
Tabla 18. Descripción de la tabla tbl_estructura .....	42
Tabla 19. Descripción de la tabla tbl_estado_horario .....	43
Tabla 20. Descripción de la tabla tbl_grupo .....	43
Tabla 21. Descripción de la tabla tbl_hora .....	43
Tabla 22. Descripción de la tabla tbl_malla .....	44
Tabla 23. Descripción de la tabla tbl_nivel .....	44
Tabla 24. Descripción de la tabla tbl_horario.....	44
Tabla 25. Descripción de la tabla tbl_mencion .....	45
Tabla 26. Descripción de la tabla tbl_materia .....	45
Tabla 27. Descripción de la tabla tbl_rango_aulas .....	46
Tabla 28. Descripción de la tabla tbl_periodo .....	46
Tabla 29. Descripción de la tabla tbl_tipo_estructura.....	46
Tabla 30. Descripción de la tabla tbl_tipo_aula .....	46
Tabla 31. Descripción de la tabla tbl_usuario .....	47
Tabla 32. Descripción de clases del paquete Persistencia .....	68

Tabla 33. Contenido de la tabla: tbl_tipo_estructura .....	76
Tabla 34. Contenido de la tabla: tbl_tipo_aula.....	76
Tabla 35. Contenido de la tabla: tbl_estadoh .....	77
Tabla 36. Contenido de la tabla: tbl_dia .....	77
Tabla 37. Contenido de la tabla: tbl_hora .....	77
Tabla 38. Contenido de la tabla: tbl_periodo .....	77
Tabla 39. Contenido de la tabla: tbl_malla .....	78
Tabla 40. Contenido de la tabla: tbl_nivel .....	78
Tabla 41. Contenido de la tabla: tbl_tipo_estructura .....	78
Tabla 42. Descripción archivo Distributivo.....	79
Tabla 43. Descripción archivo Materias Ofertadas .....	79
Tabla 44. Descripción archivo Horario .....	79
Tabla 45. Pruebas contra requerimientos.....	87

## ÍNDICE DE ANEXOS

Anexo 1. Carga Final Tbl_usuario .....	98
Anexo 2. Carga Final Tbl_estructura .....	98
Anexo 3. Carga Final Tbl_carrera .....	99
Anexo 4. Carga Final Tbl_materia .....	99
Anexo 5. Carga Final Tbl_distributivo .....	100
Anexo 6. Carga Final Tbl_horario .....	100
Anexo 7. Reporte por aula .....	101
Anexo 8. Reporte por docente .....	102
Anexo 9. Reporte por Grupo .....	103

## **RESUMEN**

El presente trabajo de titulación, realiza una descripción de la problemática actual que posee la Sede Quito en el proceso de asignación de espacios físicos, a la vez que se presenta una breve reseña de los proyectos presentados para dar solución a esta problemática y describe el Proyecto UPS Schedule al cual pertenece este trabajo.

El sistema desarrollado denominado “Gestión de Asignación de Espacios Físicos” es una aplicación de escritorio implementado mediante el uso exclusivo de herramientas y plataformas de software libre, está dirigido a los Asistentes Administrativos de los Campus que conforman la Sede Quito y permite a estos gestionar la asignación de espacios físicos a los horarios de clase y la generación de reportes detallados por Aula, Docente o Grupo de Clase.

A continuación se muestra el desarrollo del Sistema, en los cuales se ejecuta en forma secuencial el Análisis de los Requerimientos, el Diseño y Construcción del Sistema, terminando con las pruebas realizadas.

## **ABSTRACT**

This paper titration, makes a description of the current problem which has headquarters Quito in the process of assigning physical spaces, while it presents a brief overview of the projects presented to solve this problem and describes the Project Schedule UPS that owns this job.

The developed system called "Managing Assignment Physical Spaces" is a desktop application implemented through the exclusive use of tools and platforms of free software, is addressed to the Directors of Physical Spaces campuses that make Headquarters Quito and allows these manage the allocation of physical spaces to class schedules and generating detailed by Classroom Teacher or Class Group reports.

Following the development of the system is shown in which runs sequentially Requirements Analysis, Design and Construction of the System, ending with the tests

## **INTRODUCCION**

Las Instituciones Educativas, ven limitada su capacidad de ofertar una mayor oferta académica por factores como: personal docente limitado y falta de espacios físicos. El proceso de asignación de espacios físicos cuando no es realizado en forma óptima, provoca subutilización de las aulas, realizar este proceso en forma manual requiere un gran esfuerzo por parte del personal administrativo.

La variación de cursos ofertados durante cada periodo académico, deriva en realizar el proceso de asignación en forma repetitiva, adicionalmente los reportes generados son muy limitados.

Para reducir el problema de déficit de aulas se han desarrollado múltiples sistemas para realizar un proceso de asignación óptimo, sin embargo la mayoría de estos sistemas tienen un alto costo y están diseñados para trabajar con un modelo estructural de espacios físicos en específico.

La implementación de un sistema que cumpla con los requerimientos específicos de cada Institución es necesario, esto permitirá: reducir déficit de espacios físicos, automatizar el proceso y generar reportes detallados.



## **CAPÍTULO 1**

### **1. MARCO REFERENCIAL Y TEÓRICO**

#### **1.1. Planteamiento del problema**

La Universidad Politécnica Salesiana fue creada en el año 1994 en la ciudad de Cuenca, con la propuesta de brindar una educación superior humanística y politécnica, en años posteriores se crearon sedes en las ciudades de Quito y Guayaquil, gracias a la aceptación obtenida.

A través de los años, la oferta académica se ha ido incrementando de forma gradual en las diferentes sedes de la Universidad, derivando en una problemática en el proceso de asignación de espacios físicos. Este proceso se lo realiza en forma manual en cada periodo académico como se describe a continuación:

- Los Directores de Carrera comunican a la Asistente Administrativa el número de cursos a ofertar.
- La Asistente Administrativa asigna de forma única un aula a cada curso.
- La Asistente registran en un archivo de Excel la asignación realizada.

Los horarios de clases de los cursos presentan horas en las cuales no reciben clases y contienen materias que deben ser tomadas en los laboratorios, durante estas horas el aula estará asignada al grupo, sin embargo se encontrará vacía provocando una subutilización de esta.

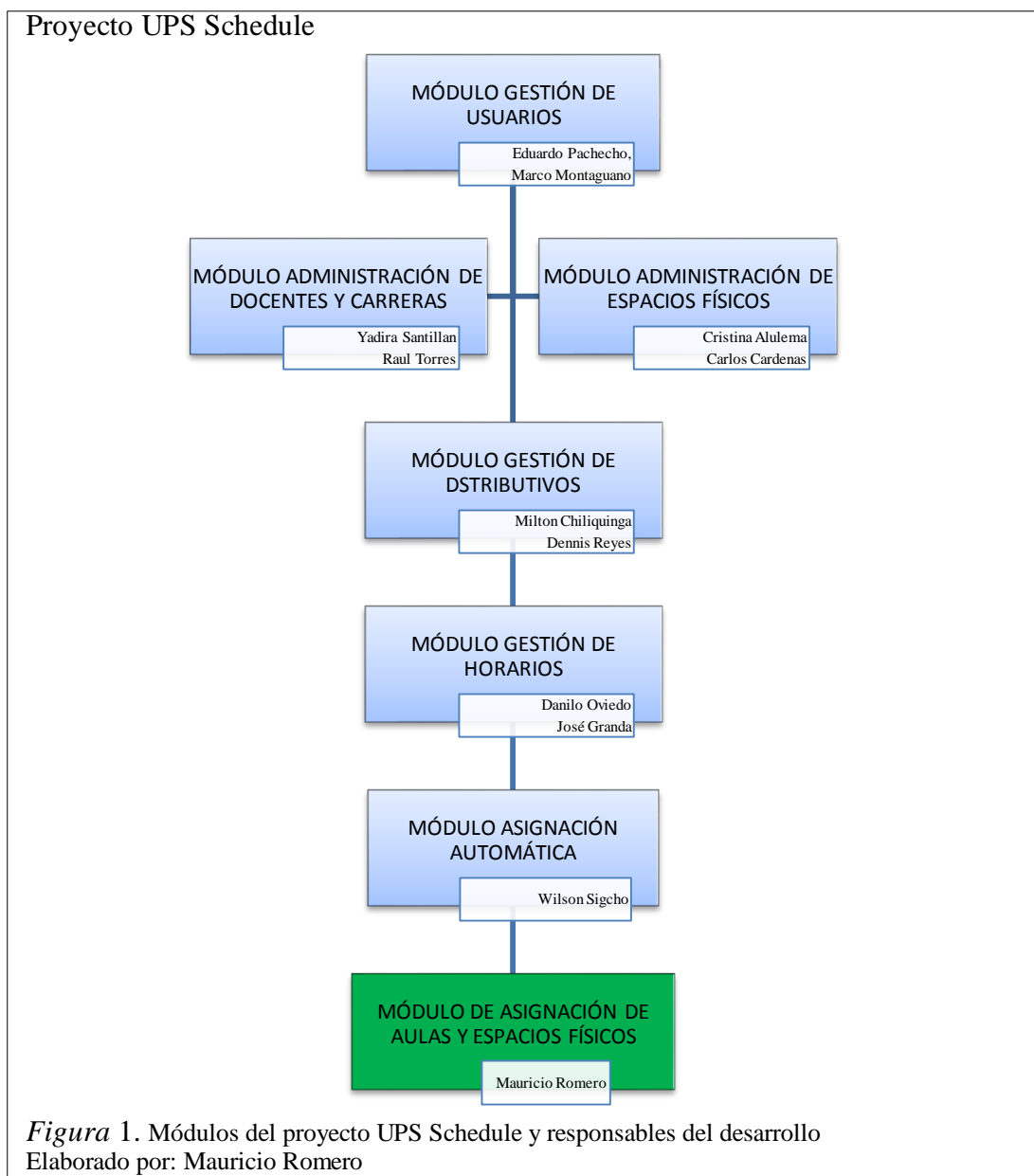
Esta problemática fue identificada y tratada durante el Periodo 37 por la Ingeniera Doris Mesa, mediante la conformación de equipos de trabajo para el desarrollo de prototipos de un sistema que permita la generación manual de horarios de clases y la asignación de aulas, este sistema estaba destinado a la Asistente Administrativa.

Posterior a un proceso de evaluación, se presentaron los mejores prototipos a las autoridades de la Universidad, quienes manifestaron que el sistema debía enfocarse en las tareas que ejecuta la Asistente Administrativo y que la gestión de la información requerida para la asignación de aulas demandaría un esfuerzo mucho mayor, por lo tanto la implementación del Sistema no fue viable.

Durante el Periodo 40, el Ingeniero Rodrigo Tufiño, realiza un nuevo análisis al proceso de asignación de espacios físicos y presenta una propuesta denominada Proyecto UPS Schedule. Este proyecto contempla el desarrollo de múltiples módulos, en los cuales se administra y gestiona la información requerida (Espacios Físicos, Carreras, Docentes, Horarios, etc.) para tratar esta problemática, tomando en consideración que los usuarios cumplirán con las mismas tareas que desempeñaban previamente, la asistente administrativa realizará la Gestión de Espacios Físicos, mientras que los Directores de Carrera se encargarán de la Gestión de Carreras, Distributivos y Horarios.

De igual manera, tomando en consideración lo lento y poco efectivo que resulta el proceso de asignación de aulas en forma manual, se incluye un módulo para asignar los espacios físicos a cada horario de forma automática mediante la implementación de un algoritmo metaheurístico.

En el Diagrama 1, muestra los módulos que integran el Proyecto UPS Schedule, además de los responsables del desarrollo de cada uno.



El proyecto contempla el desarrollo de los siguientes módulos:

- **Módulo Gestión de Usuarios:** creación de usuarios y su administración de forma segura y auditada. Está dirigido a los Administradores Informáticos. (Marco Montaguano, 2014)

- Módulo de Administración de Docentes y Carreras: administración de las carreras y la disponibilidad de los Docentes. Está dirigido a los Administradores Informáticos y a los Docentes. (Yadira Santillan, 2014)
- Módulo Administración de Espacios Físicos: administración de los diferentes tipos de espacios físicos con los que cuenta la UPS. Está dirigido a los Administradores Informáticos. (Cristina Alulema, 2014)
- Módulo Gestión de Distributivos: gestionar los Docentes y sus distributivos. Está dirigido a los Directores de Carrera. (Milton Chiliquinga, 2014)
- Módulo de Gestión de Horarios: gestionar los horarios de los diferentes campus. Está dirigido a los Directores de Carrera.
- Módulo Asignación Automática: mediante el uso de Inteligencia Artificial, este módulo realiza la asignación automática de espacios físicos a cada horario. Está dirigido a los Administradores Informáticos.
- Módulo Asignación Automática: mediante el uso de Inteligencia Artificial, este módulo realiza la asignación automática de espacios físicos a cada horario. Está dirigido a los Administradores Informáticos.
- Módulo de Asignación de Aulas y Espacios Físicos: este módulo es tratado en el presente trabajo, y permite:
  - La asignación obtenida por el uso del algoritmo metaheurístico.
  - La asignación de espacios físicos a los horarios que no fueron asignados por el algoritmo.
  - Obtener los reportes del trabajo realizado.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

- Obtener un módulo funcional para la Gestión de la Asignación de Espacios Físicos que trabaje de forma integrada con los demás módulos del Proyecto UPS Schedule para la Universidad Politécnica Salesiana Sede Quito.

### **1.2.2. Objetivos específicos**

- Analizar el proceso de asignación de espacios físicos, para identificar las tareas que cumple la Asistente Administrativa del Campus.

- Integrar el módulo de gestión de aulas y espacios físicos al sistema UPS Schedule.
- Generar reportes de la Asignación de espacios físicos, para mostrar a detalle el uso de cada instalación física asignada.
- Implementar procedimientos de seguridad que garanticen la integridad del Sistema y de la información almacenada en el Base de datos.
- Ejecutar pruebas contra requerimientos, para evaluar el cumplimiento de estos.

### **1.3. Justificación**

El Proyecto UPS Schedule conlleva el desarrollo de múltiples módulos, el presente trabajo de titulación se refiere al módulo denominado módulo de Asignación de Aulas y Espacios Físicos, el cual desarrollará un sistema dirigido a los Asistentes Administrativos de los Campus, con la finalidad de brindar una herramienta para gestionar la asignación realizada por el módulo de Asignación Automática o para realizar la asignación manual.

La implementación de este módulo permitiría además la generación automática de Reportes en los cuales se brindará información detallada de la asignación realizada, los reportes están orientados a mostrar los horarios y la asignación de espacios por aula, docente y grupo de clases.

Actualmente la Universidad no posee un sistema en el cual se pueda conocer el estado de la asignación, el desarrollo de este módulo contempla el desarrollo de una interfaz en la cual se podrá obtener esta información por medio de cuadros estadísticos, en esta interfaz se visualizará el porcentaje total de horas con o sin asignación, además de mostrar el estado de la asignación por cada carrera al igual que el uso realizado a cada bloque del campus.

### **1.4. Alcance**

Las tareas que serán realizadas son detalladas a continuación:

- Asignar, modificar y eliminar asignaciones de Espacios Físicos.

- Preestablecer aulas a en las cuales se podrá asignar una carrera, mismas que serán consideradas por el módulo de Asignación Automática en su ejecución.
- Identificar horas de clases que no tienen un aula asignada.
- Generar reportes por Aula, Docente y Grupo, en los cuales se detallará:
  - Reporte Aula: mostrará los grupos y materias asignados en cada hora del aula.
  - Reporte Docente: mostrará el aula en la cual el docente impartirá cada hora de clase.
  - Reporte Grupo: mostrará las aulas en las cuales los grupos recibirán sus clases.

Este módulo no contempla la gestión o administración de: espacios físicos, horarios, docentes, grupos, carreras, etc., estos procesos son realizados por otros módulos del sistema, como ha sido detallado previamente.

## **1.5. Marco Teórico**

### **1.5.1. Programación orientada a objetos**

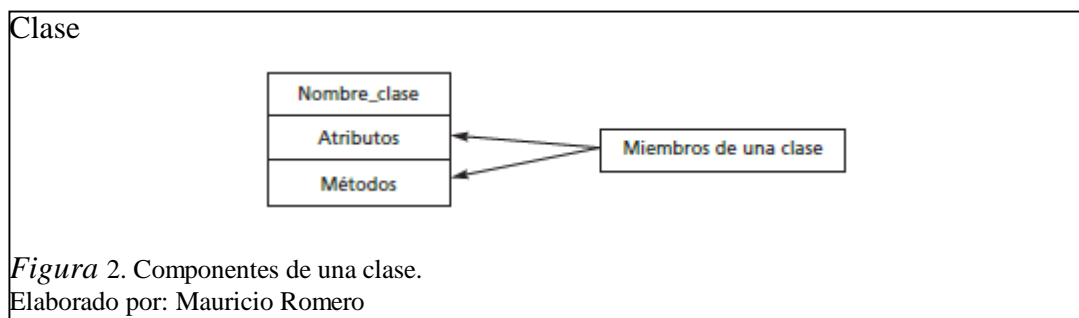
La POO viene de la evolución de la programación estructurada; básicamente la POO simplifica la programación con la nueva filosofía y nuevos conceptos que tiene. La POO se basa en dividir el programa en pequeñas unidades lógicas de código. A estas pequeñas unidades lógicas de código se les llama objetos. Los objetos son unidades independientes que se comunican entre ellos mediante mensajes. (Ciberaula España)

“Los objetivos de esta modalidad de programación son el mejoramiento de la productividad del programador por medio del manejo de la complejidad del software a través del uso de clases y sus objetos asociados”. (Barraza, 2006, pág. 10)

Los elementos de la POO son:

## Clase

El elemento clave de la programación orientada a objetos es la clase. Una **clase** se puede definir como una descripción abstracta de un grupo de objetos, cada uno de los cuales se diferencia por su *estado* específico y por la posibilidad de realizar una serie de *operaciones*. Por ejemplo, una pluma estilográfica es un objeto que tiene un estado (llena de tinta o vacía) y sobre la cual se pueden realizar algunas operaciones (por ejemplo escribir, poner o quitar el capuchón, llenar de tinta si está vacía) (Aguilar, 1996, pág. 17)



La figura 1. Muestra los componentes de una clase: atributos y métodos.

- **Atributos:** son las características generales que describen un grupo de Objetos. Un atributo consta de tres partes: tipo de dato, nombre del atributo y el valor del atributo. Por Ejemplo: String Nombre = “Juan”.
- **Métodos:** “Un método (función) es un conjunto de instrucciones o sentencias que realiza una determinada tarea; se identifica con un nombre y puede o no devolver un valor”. (Barraza, 2006, pág. 44).

## Objetos

Un Objeto es una instancia de una clase, es decir, posee los atributos y métodos de una clase, considere a los seres humanos como un grupo de objetos que se pueden definir en la clase Ser Humano, cada persona sería una instancia de esta clase. Compartirían atributos con igual o diferente valor como nombre, peso, estatura, edad, color de ojos etc.

## **Mensajes**

Un Mensaje es la acción de efectuar una llamada a un método, instruyéndole de llevar a cabo una acción, estos mensajes pueden incluir o no parámetros. Para una mejor comprensión considere a 2 personas como objetos de la clase ser humano las cuales poseen los métodos: preguntar y responder, el Objeto A ejecuta el método preguntar invocando al método responder del Objeto B, el mensaje puede enviar distintos parámetros como nombre, edad, peso, sexo, etc. Que serán interpretados por el Objeto B y determinaran su respuesta.

## **Métodos Especiales**

“Son métodos cuyo objetivo es asignar espacio de memoria a un objeto o liberar espacio de memoria asignado al objeto, los cuales se implementan a través de los métodos llamados Constructores y Destructores”. (Barraza, 2006, pág. 53)

- **Constructor**

Es un método para construir un nuevo objeto y asignar valores iniciales a sus miembros datos. El propósito de un constructor es garantizar que el objeto permanezca en un estado consistente en el momento de su creación. Al no existir un constructor, el compilador genera uno automáticamente. (Barraza, 2006, pág. 53)

- **Destructor:**

Es un método que libera la memoria asignada a un objeto cuando este fue inicializado.

Las principales características de la POO son:

## **Abstracción**

Es la capacidad de crear tipos de datos definidos por el usuario. Es una definición de tipo que incluye datos y funciones (métodos) en una sola estructura definida llamada clase. La abstracción de datos es un



concepto fundamental en el diseño de un programa, pues permite definir el dominio y la estructura de los datos (atributos), junto con una colección o conjunto de operaciones que tienen acceso a los mismos (métodos). La abstracción permite no preocuparse de los detalles accesorios, y generalizar y centrarse en los aspectos que permiten tener una visión global del problema. (Barraza, 2006, pág. 10)

## Herencia

Propiedad que permite a los objetos ser contruidos a partir de otros objetos. Dicho de otro modo, la capacidad de un objeto para utilizar las estructuras de datos y los métodos previstos en antepasados o ascendientes. El objetivo final es la reutilizabilidad o reutilización, es decir reutilizar código anteriormente ya desarrollado. (Aguilar, 1996, pág. 84)



## Polimorfismo

Esta característica es crítica porque permite que varios tipos (derivados de un mismo tipo base) sean tratados como si fueran uno sólo, y un

único fragmento de código se puede ejecutar de igual forma en todos los tipos diferentes. La llamada a un método polimórfico permite que un tipo exprese su distinción de otro tipo similar, puesto que ambos se derivan del mismo tipo base. Esta distinción se expresa a través de diferencias en comportamiento de los métodos a los que se puede invocar a través de la clase base. (Eckel, 2002, pág. 223)

### **1.5.2. Java**

“Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems”. (Sun Microsystem, 2015)

Java está orientado a objetos y posee las siguientes características:

#### **Simple**

Posee una sintaxis simplificada de C++.

Las principales simplificaciones consisten en no necesitar tener ficheros de cabeceras, la ausencia de punteros, no tener que gestionar la memoria y disponer de un extenso surtido de librerías que nos facilita la faena cuando tenemos que enfrentarnos a programas de cierta complejidad. También tenemos muchas menos estructuras de programación y habitualmente más sencillas. (Garcia, 2003, pág. 9)

#### **Orientado a Objetos**

Java está basado en C++, pero a diferencia de este, Java se encuentra orientado completamente al uso de objetos. En java todo, a excepción de las variables fundamentales es un objeto.

## **Distribuido**

Una de las líneas principales que marcaron el diseño de Java fue su orientación a usos relacionados con Internet. Cualquiera que haya intentado programar alguna aplicación Internet en otro lenguaje se dará cuenta de la simplicidad de Java en estos menesteres. Incluso la programación de CGI's resulta mucho más simple. También es bastante común la ejecución de parte (o toda) la aplicación por parte del cliente, con los llamados applets. (Garcia, 2003, pág. 9)

## **Arquitectura neutral**

El código compilado de Java, puede ser ejecutado en una gran variedad de Sistemas Operativos de Software, esto es posible gracias a la implementación de la JVM que interpreta el código y lo traduce a cualquier lenguaje de máquina.

## **Interpretado**

El bytecode generado por el compilador siempre es el mismo independientemente de la plataforma en la que es ejecutado.

## **Multiflujo**

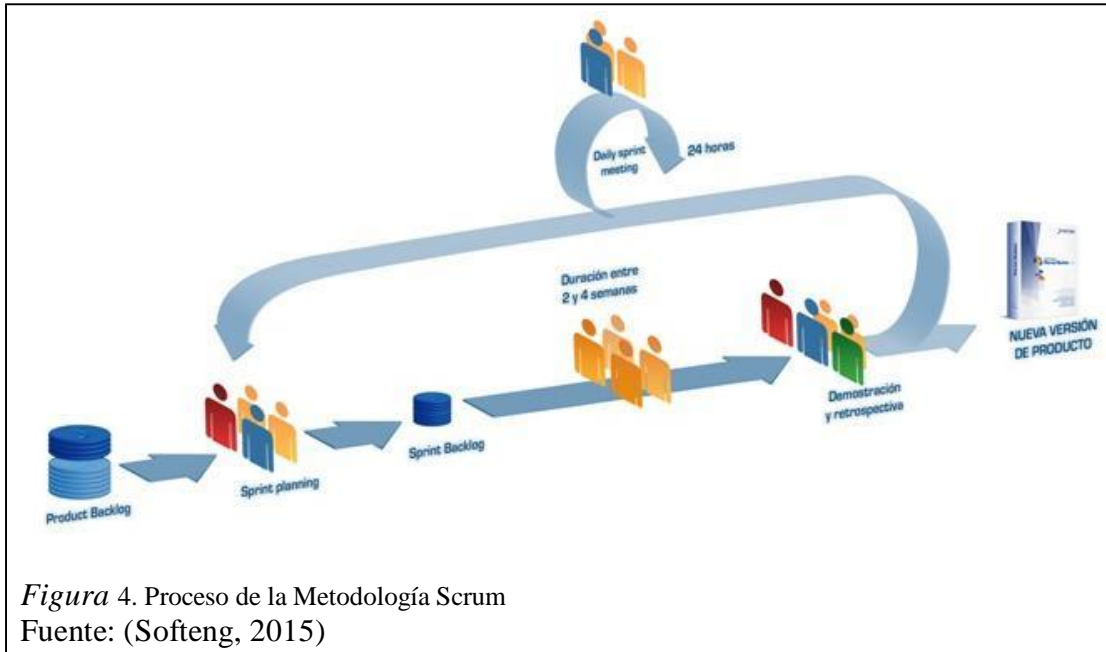
Soporta en forma nativa el uso de threads sin el uso de librerías específicas. Esto permite aprovechar las ventajas de los sistemas multiprocesadores.

### **1.5.3. Metodología SCRUM**

SCRUM es una de las más conocidas metodologías ágiles para la gestión de proyectos. Las metodologías ágiles se centran en aspectos como la flexibilidad en la introducción de cambios y nuevos requisitos durante el proyecto, el factor humano, el producto final, la colaboración con el cliente y el desarrollo incremental como formas de asegurar los

buenos resultados en proyectos con requisitos muy cambiantes o cuando se exige, como es habitual, reducir los tiempos de desarrollo manteniendo una alta calidad. (Proyectalis, 2015)

#### Proceso de la metodología SCRUM



*Figura 4.* Proceso de la Metodología Scrum  
Fuente: (Softeng, 2015)

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos del producto, que realiza durante la iteración con vista a las siguientes iteraciones. (Proyectos Agiles, 2013)

La metodología SCRUM brinda las siguientes ventajas:

- Es fácil de aprender.
- Requiere muy poco esfuerzo para comenzarse a utilizar.
- Permite que abarcar proyectos donde los requisitos de negocio están incompletos
- Permite el desarrollo, testeo y correcciones rápido
- Mediante las reuniones diarias se ven claramente los avances y problemas
- Como toda metodología ágil, obtiene mucho feedback del cliente.
- Facilita la entrega de productos de calidad a tiempo.

## **CAPÍTULO 2**

### **2. ANÁLISIS Y DISEÑO**

En este capítulo se presenta la solución propuesta, los requerimientos del sistema y el análisis realizado para el proyecto.

#### **2.1. Análisis de viabilidad**

El presente análisis considera los aspectos fundamentales para el desarrollo de un proyecto de software.

##### **2.1.1. Viabilidad técnica**

El sistema ha sido elaborado para funcionar en un ambiente de escritorio, permitiendo un acceso rápido a la información contenida en la base de datos.

Las funciones de este sistema están orientadas a un usuario único por campus, motivo por el cual no se requiere la implementación de este en un ambiente web.

La Universidad Politécnica Salesiana Sede Quito, cuenta con el hardware necesario para la implementación del presente sistema.

Características:

Gabinete para servidores Blades, con los siguientes componentes:

- 1 Enclosure Blade c3000, con conexión eléctrica bifásica.
- Switches de LAN 1 GB E2 para Enclosure C3000.
- Sistema de ventiladores (4 instaladas) y fuentes redundantes (2 instaladas).
- 1 Módulo On Board Administración.

Servidor para aplicaciones, Servidor BL460C:

- 1 Procesador Intel Xeon Quad Core, 2.33 GHz 1x4MB cache.
- 4 GB de Memoria RAM.
- 1 Smart Array RAID 1.

- Discos de 146GB 10k SAS HDD

Equipo para desarrollo por parte del tesista:

Laptop Lenovo Ideapad:

- Procesador: intel core I7
- Memoria: 8 GB
- Disco: 500 GB SSD
- Video: 4 GB

### 2.1.2. Viabilidad operativa

Con la experiencia obtenida durante el periodo 37, el proyecto UPS Schedule se definió en múltiples módulos que cumplen las funciones que actualmente realizan los usuarios a los cuales están dirigidos, el presente trabajo está enfocado a automatizar las tareas de asignación que realizan los Administradores de los Espacios Físicos de las sede Quito, por lo tanto se espera que el sistema sea aceptado y utilizados por estos.

### 2.1.3. Viabilidad económica

Al ser parte de un proyecto de titulación, los gastos de ANÁLISIS, DISEÑO, DESARROLLO E INTEGRACIÓN del presente proyecto serán asumidos por el Tesista.

El Sistema no requiere la implementación de un nuevo hardware, por lo que no representaría gastos para la Institución.

|

Tabla 1. *Costos de Implementación*

Concepto	Valor	Responsable
Software		
Netbeans IDE 8.0	\$0	Tesista
Jcommon 1.0.21	\$0	
Itexpdf 5.5.2	\$0	
Jfreechart 1.0.17	\$0	
PostgreSQL 9.3	\$0	
SQL Power Architect 1.05	\$0	
Libre Office 4	\$0	
Pencil	\$0	
Hardware		

1 Laptop	\$1400	Tesista
1 Impresora	\$120	
Talento Humano Desarrollador	\$2400	Tesista
Materiales de Oficina	\$10	Tesista
Resmas de papel bond	\$8	
Copias		
Gastos Adicionales		Tesista
Internet	\$420	
Matrícula	\$330	
Derechos de Tutor	\$400	
Derecho de certificación	\$40	
Alimentación	\$360	
Transporte	\$300	
<b>Total</b>	<b>\$6092</b>	

Nota.

Elaborado por: Mauricio Romero

## 2.2. Análisis de requerimientos

En esta sección se describe detalladamente los requerimientos de software para el diseño del programa “Gestión de espacios físicos” que será implementado en la Universidad Politécnica Salesiana – Sede Quito, este documento está dirigido al personal técnico que se encargará de desarrollar el sistema.

En los apartados siguientes se definen las características y funcionalidades que deberá tener el sistema “Gestión de espacios físicos”.

### 2.2.1. Definiciones y abreviaturas

- **UPS:** abreviación de Universidad Politécnica Salesiana.
- **Periodo:** numeración secuencial de semestres en los que ha operado la UPS.
- **Estructura:** instalación física de propiedad de la UPS.
- **Tipo de estructura:** categorización a las Instalaciones de la UPS: P ej. Sede, Campus, Aula, etc.
- **Tipo de aula:** categorización a las Instalaciones de tipo Aula: P ej. Regular, Laboratorio, etc.
- **Mención:** especialización en el plan de estudios de una Carrera: P ej. Ingeniería de Sistemas – Gestión para la Informática, Telemática.
- **Grupo:** código que identifica el paralelo de un nivel perteneciente a una carrera.
- **Horario:** espacio de tiempo que hace referencia a un día y hora preestablecido para impartir una materia.



- **Horario asignado:** horario que posee un aula asignada.
- **Horario sin asignación:** horario que no cuenta con un aula asignada.
- **Aula disponible:** aula sin asignación en una determina hora y día.
- **Aula ocupada:** aula que tiene un horario asignado.
- **Bloque:** espacio físico que contiene una o más aulas.

### 2.2.2. Descripción general

### 2.2.3. Perspectiva del producto

El sistema Gestión de espacios físicos será parte del Sistema UPS Schedule y está orientado a ser utilizado por los Administradores de los Campus, para su correcto funcionamiento dependerá de la información ingresada a la Base de Datos por los módulos de Gestión de Horarios, Carreras y Usuarios, del módulo de Administración de Espacios Físicos y del módulo de Asignación Automática.

### 2.2.4. Funciones del producto

- **Visualización de estadísticas**

Esta sección mostrará el estado de la Asignación de Espacios, mediante cuadros estadísticos.

Para empezar, se observará un cuadro con el porcentaje de la asignación total realizada; a continuación, se observará el estado de la asignación por carrera; para finalizar, se observará el estado de la asignación por bloque.

La información que se muestre, será la relacionada al usuario que ingresó al sistema.

- **Asignación de materias pendientes**

Este submódulo permitirá visualizar, seleccionar y asignar un aula a los horarios de materias que el algoritmo metahurístico no asigne de forma automática, para realizar esta tarea de forma rápida, se desplegará un listado de los espacios físicos disponibles que cumplan con los requerimientos de la materia.

Adicionalmente cuando se seleccione un aula, se mostrará una vista previa de los horarios asignados a esta y la asignación a realizar.

- **Gestión de espacios físicos**

En este submódulo se podrá visualizar todos los espacios físicos con los que cuenta la institución, a la vez que se mostrará en forma de grilla todas las materias que ocupan estos espacios en los diferentes días y horas.

También se incluirá la funcionalidad de eliminar asignaciones de materias en los diferentes espacios.

- **Reportes**

La función de este submódulo será la de exportar a un documento PDF la información de los horarios y aulas asignadas.

Se contempla el desarrollo de 3 reportes:

- Reporte de aula
- Reporte de docente
- Reporte de grupo

El reporte de Aula se exportará a un archivo PDF con la información de las materias y grupos que se encuentra asignado a cada Aula.

El reporte de Docente se exportará a un archivo PDF con la información del aula asignada a cada materia que dicta el Docente.

El reporte de Grupo se exportará a un archivo PDF con la información de la asignación de aula realizada a cada materia que recibe el grupo.

Cabe destacar, que para la generación de cada reporte se podrá aplicar distintos filtros.

### **2.2.5. Características de usuarios**

El uso del sistema está limitado a los Asistentes Administrativos de los campus de la UPS – Sede Quito, puesto que son usuarios con experiencia en el manejo de los espacios físicos y tienen un perfil técnico adecuado.

### 2.2.6. Suposiciones y dependencias

Este documento de requerimientos, está basado en las siguientes condiciones:

- Estructura actual de los espacios físicos y de las carreras.
- Horario de uso de las aulas desde las 7.00 hasta las 21.30, de lunes a domingo.
- Receso de 30 minutos de 13.00 a 13.30.
- La duración de la hora de clase está considerada de 60 minutos.

### 2.2.7. Requisitos futuros

Se puede implementar la integración con el Sistema Nacional Académico (SNA), para la descarga automática de la información que requiere el sistema “Gestión de espacios físicos”.

### 2.2.8. Requisitos específicos

La definición de los requerimientos funcionales, se realizó utilizando la plantilla organizada por clase de usuarios (Especificación de requerimientos de software IEEE Std 830-1998)

#### 2.2.8.1. Acceder al sistema

Tabla 2. *Requisito funcional 1*

Descripción	Limitar el uso del aplicativo, será utilizado por los usuarios administradores de los campus
Precondición	Usuarios, Estructuras en la base de datos
Entrada	Usuario y contraseña
Proceso	Validación de usuario y contraseña
Salida	Acceso al Sistema

Nota.

Elaborado por: Mauricio Romero

#### 2.2.8.2. Modificar clave de acceso

Tabla 3. *Requisito funcional 2*

Descripción	El sistema debe permitir el cambio de contraseña
Precondición	Usuario Autenticado
Entrada	Contraseña Actual – nueva contraseña – confirmación nueva contraseña
Proceso	Validar contraseña actual – validar contraseña y confirmación
Salida	Mensaje confirmación de cambio de contraseña

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.3. Visualizar estadísticas de asignación

Tabla 4. *Requisito funcional 3*

Descripción	Se debe visualizar estadísticas del estado actual de la asignación: por campus, carrera y bloque
Precondición	Datos existentes en la BDD.
Entrada	
Proceso	
Salida	Gráficos estadísticos del estado de la asignación

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.4. Gestión materias sin aula

Tabla 5. *Requisito funcional 4*

Descripción	El sistema debe permitir una fácil asignación de aulas a las materias que no fueron asignadas por el algoritmo inteligente
Precondición	Materias sin aulas
Entrada	Horario sin asignación – Aula disponible
Proceso	Asignar aula al horario
Salida	Mensaje de confirmación

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.5. Eliminar asignación de aula

Tabla 6. *Requisito funcional 5*

Descripción	El sistema debe permitir la eliminación de materias en las aulas
Precondición	Asignación existente
Entrada	Aula – Horarios a eliminar
Proceso	Eliminar la asignación del aula en los horarios
Salida	Mensaje de confirmación

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.6. Generación de reporte por docente

Tabla 7. *Requisito funcional 6*

Descripción	El sistema debe permitir la generación de reporte por docente
Precondición	Horarios existentes
Entrada	Selección de carrera
Proceso	Generación de reporte
Salida	Archivo PDF

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.7. Generación de reporte por grupo

Tabla 8. *Requisito funcional 7*

Descripción	El sistema debe permitir la generación de reporte por grupo
Precondición	Horarios existentes
Entrada	Selección de carrera – nivel – grupo
Proceso	Generación de reporte
Salida	Archivo PDF

Nota.

Elaborado por: Mauricio Romero

### 2.2.8.8. Generación de reporte por aula

Tabla 9. *Requisito funcional 8*

Descripción	El sistema debe permitir la generación de reporte por aula
Precondición	Horarios – Asignación existentes
Entrada	Selección de Bloque – piso – aula
Proceso	Generación de reporte
Salida	Archivo PDF

Nota.

Elaborado por: Mauricio Romero

### 2.2.9. Atributos del sistema

La seguridad del sistema estará dada por el acceso restringido al Administrador de Espacios Físicos de cada campus, mediante un usuario y contraseña. El desarrollo del mismo se realizará en lenguaje JAVA, permitiendo su uso en cualquier sistema operativo que soporte JAVA.

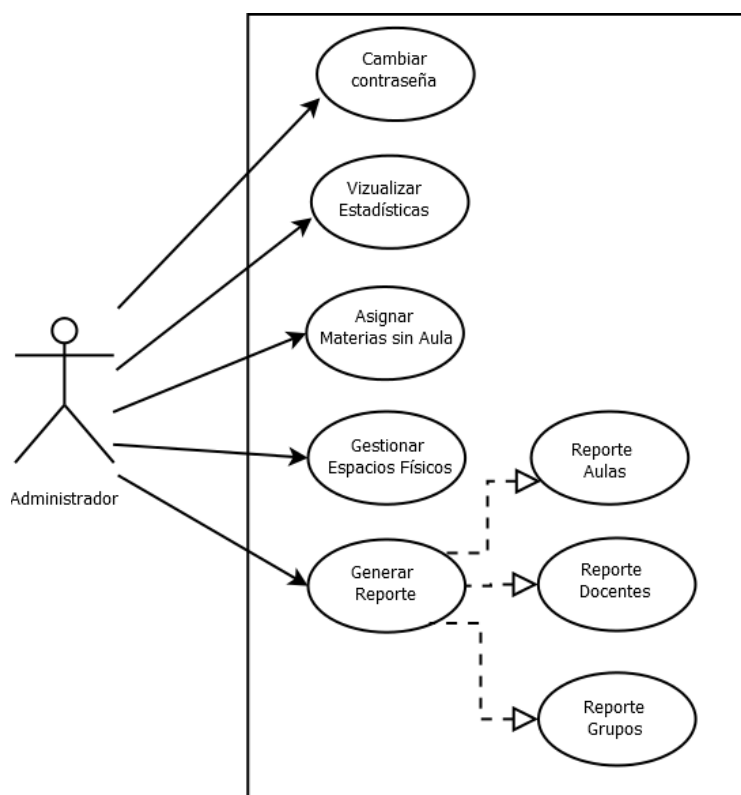
Las credenciales para la comunicación al Servidor de BDD, serán proporcionadas por un archivo de configuración, en los cuales constará la dirección del servidor, puerto de comunicación, usuario, contraseña y el nombre de la BDD.

## 2.3. Diseño

En esta sección se presenta la solución planteada para la elaboración del programa Gestión de Espacios Físicos, se definen las tareas que podrá realizar el usuario y la estructura con la que contará el sistema

### 2.3.1. Diagrama de casos de uso

En el siguiente diagrama se muestran las principales tareas que podrá realizar el Administrador de Espacios Físicos de cada Campus.



*Figura 5.* Diagrama de Casos de Uso.

Elaborado por: Mauricio Romero

*Tabla 10. Descripción caso de uso cambiar contraseña*

Nombre:	Cambiar Contraseña
Autor:	Mauricio Romero
Fecha:	10-03-2015
Descripción: Permite cambiar la contraseña con la cual se autentifica el usuario.	
Actores: Administrador	
Precondiciones: Usuario Autenticado	
Flujo Normal: 1. Usuario ingresa en l sub menú de cambio de contraseña. 2. Usuario ingresa contraseña actual. 3. Usuario ingresa nueva contraseña. 4. Usuario ingresa confirmación de nueva contraseña. 5. Usuario presiona botón “Guardar”. 6. Se muestra mensaje de confirmación para realizar el cambio. 7. Usuario confirma que desea realizar el cambio de contraseña. 8. Sistema valida información correcta y los almacena.	

Flujo Alternativo: 5. Usuario presiona botón “Cancelar” 7. Usuario cancela la operación 8. Sistema valida información correcta, en caso de datos erróneos, se despliega mensaje y permite modificación.
Poscondiciones: Se muestra mensaje de cambio realizado

Nota.

Elaborado por: Mauricio Romero

Tabla 11. *Descripción caso de uso visualizar estadísticas*

Nombre:	Visualizar Estadísticas
Autor:	Mauricio Romero
Fecha:	10-03-2015
Descripción:	Permite ver cuadros estadísticos sobre el estado de la Asignación de Espacios Físicos
Actores:	Administrador
Precondiciones:	Usuario Autenticado
Flujo Normal:	1. Usuario ingresa a la pestaña Home.
Flujo Alternativo:	
Poscondiciones:	

Nota.

Elaborado por: Mauricio Romero

Tabla 12. *Descripción caso de uso asignar materias sin aula*

Nombre:	Asignar Materias sin Aula
Autor:	Mauricio Romero
Fecha:	10-03-2015
Descripción:	Permite asignar de forma manual la asignación de un aula a una materia.
Actores:	Administrador
Precondiciones:	Usuario Autenticado
Flujo Normal:	1. Usuario ingresa a la pestaña Materias sin Asignar. 2. Usuario selecciona Carrera – Nivel – Grupo – Materia – Horario/s a asignar 3. Sistema busca aulas disponibles y las muestra. 4. Usuario selecciona aula a asignar. 5. Usuario presiona botón de guardar. 6. Sistema muestra mensaje de confirmación para realizar la asignación. 7. Usuario confirma asignación. 8. Sistema almacena asignación.
Flujo Alternativo:	3. Sistema busca aulas disponibles, en el caso de no encontrar ninguna, muestra mensaje de error. 7. Usuario cancela asignación.
Poscondiciones:	Se muestra mensaje de asignación realizada Sistema actualiza lista de Materias sin Aula

Nota.

Elaborado por: Mauricio Romero

Tabla 13. *Descripción caso de uso gestionar espacios físicos*

Nombre:	Gestionar Espacios Físicos
Autor:	Mauricio Romero
Fecha:	10-03-2015
Descripción:	Permite visualizar y eliminar la asignación que contiene cada espacio físico.
Actores:	Administrador
Precondiciones:	Usuario Autenticado
Flujo Normal:	<ol style="list-style-type: none"> <li>1. Usuario ingresa a la pestaña Espacios Físicos.</li> <li>2. Usuario selecciona Bloque – Piso – Aula a visualizar o modificar.</li> <li>3. Sistema muestra la asignación que contiene el aula seleccionada.</li> <li>4. Usuario presiona botón “Editar”.</li> <li>5. Usuario selecciona asignaciones a eliminar.</li> <li>6. Sistema muestra contador de registros a eliminar.</li> <li>7. Usuario presiona botón “Guardar”.</li> <li>8. Sistema muestra mensaje de confirmación de edición.</li> <li>9. Usuario confirma edición.</li> <li>10. Sistema elimina asignaciones.</li> </ol>
Flujo Alternativo:	<ol style="list-style-type: none"> <li>5. Usuario selecciona asignaciones a eliminar, en el caso de que la Edición no esté habilitada, sistema muestra mensaje solicitando que se active la Edición presionando el botón “Editar”.</li> <li>10. Usuario cancela edición.</li> </ol>
Poscondiciones:	<p>Sistema muestra mensaje de asignación eliminada.</p> <p>Sistema actualiza lista de Materias sin Aulas.</p>

Nota.

Elaborado por: Mauricio Romero

Tabla 14. *Descripción caso de uso generar reporte*

Nombre:	Generar Reporte
Autor:	Mauricio Romero
Fecha:	10-03-2015
Descripción:	Permite generar reportes en formato PDF de la Asignación dada por Aula, Docente y Grupo
Actores:	Administrador
Precondiciones:	Usuario Autenticado
Flujo Normal:	<ol style="list-style-type: none"> <li>1. Usuario selecciona al sub menú Reportes.</li> <li>2. Usuario selecciona tipo de reporte AULA – DOCENTE - GRUPO.</li> <li>3. Sistema muestra ventana para generar el reporte seleccionado.</li> <li>4. Usuario selecciona filtros para la generación del reporte en caso de que requiera</li> <li>5. Usuario presiona botón “Generar”</li> <li>6. Sistema muestra ventana para seleccionar ruta y nombre del archivo a generar.</li> <li>7. Usuario ingresa ruta y nombre del archivo.</li> <li>8. Sistema valida ruta y nombre del archivo y lo genera.</li> <li>9. Sistema abre el reporte generado</li> </ol>
Flujo Alternativo:	<ol style="list-style-type: none"> <li>4. En el caso del que usuario no seleccione ningún filtro, el reporte se generará con toda la información.</li> </ol>



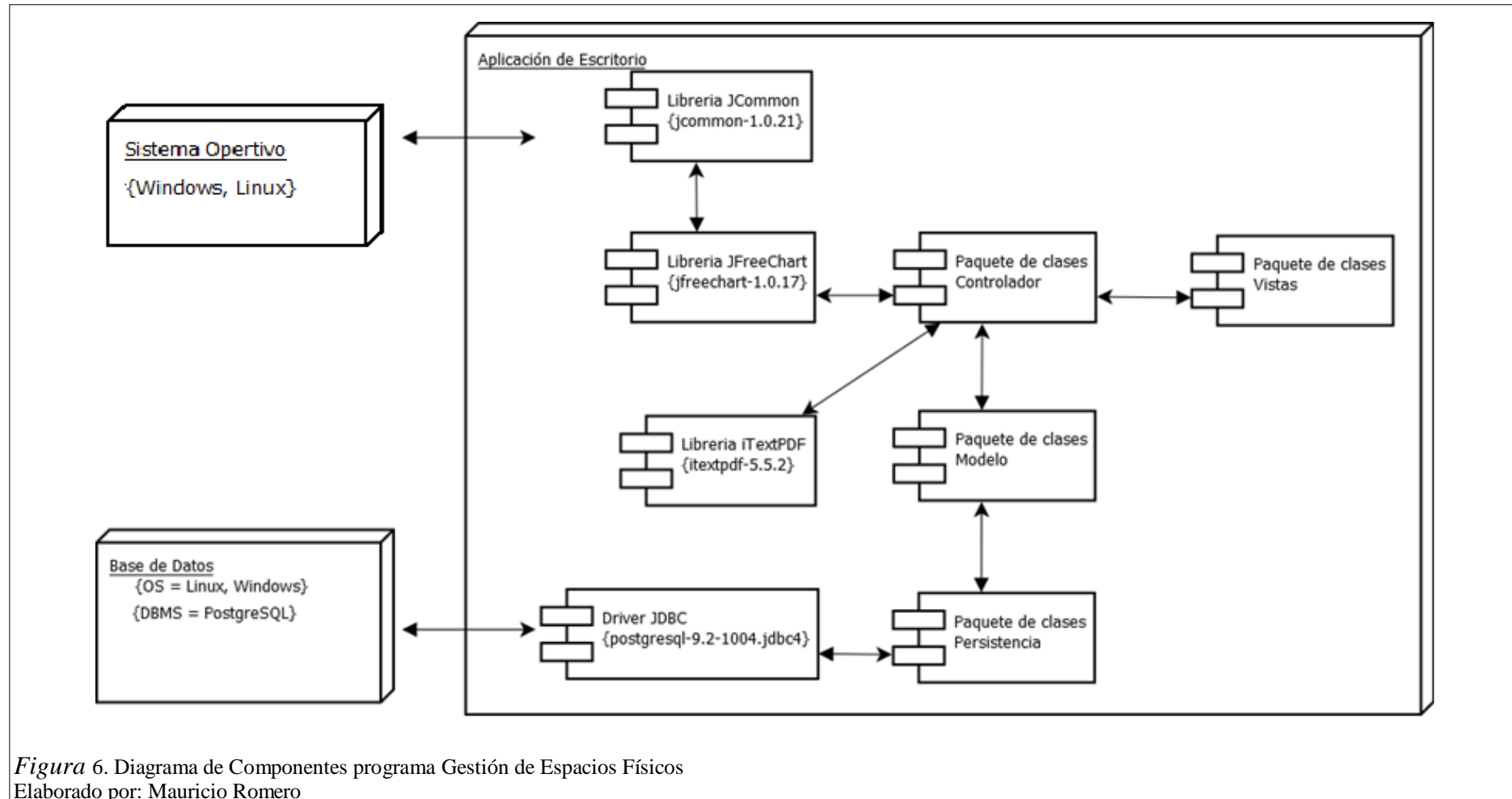
5. Sistema valida ruta y nombre, en el caso de existir algún error, se muestra mensaje de error y se solicita cambio de ruta o nombre.
--

Poscondiciones:
-----------------

Nota.

Elaborado por: Mauricio Romero

### 2.3.2. Diagrama de componentes



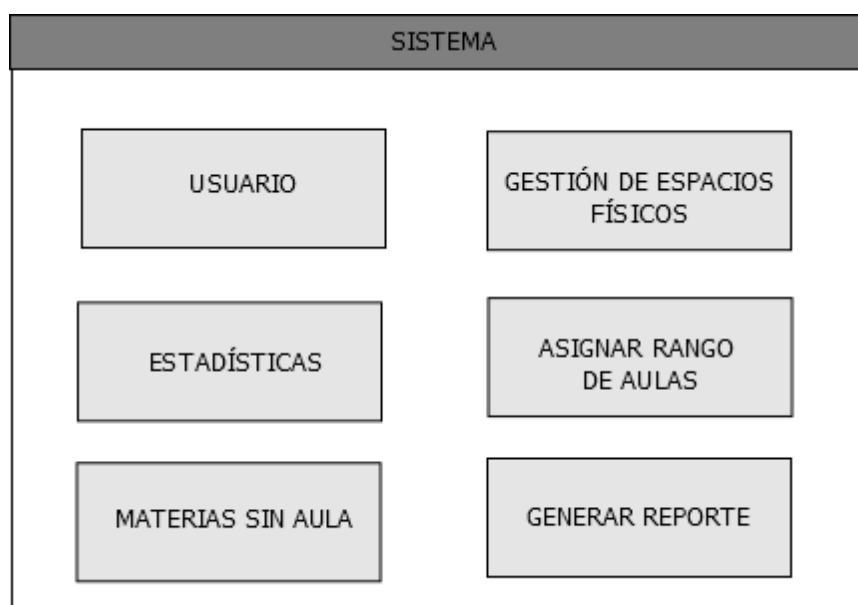
Este diagrama muestra la interrelación que tendrán los diferentes componentes del programa “Gestión de Espacios Físicos”.

La Base de Datos puede ser instalada en los Sistemas Operativos Linux y Windows y el DBMS será PosrgreSQL.

La aplicación se conectará al Motor de Base de Datos mediante Driver JDBC y podrá ser ejecutada en Sistemas Operativos Linux y Windows.

La aplicación utilizará las librerías ITextPdf, JFreeChart y JCommon.

### 2.3.3. Diagrama de bloques



*Figura 7.* Diagrama de Bloques programa Gestión de Espacios Físicos  
Elaborado por: Mauricio Romero

Este diagrama muestra los submódulos funcionales que contendrá el sistema para implementar los requerimientos establecidos en la Especificación de Requerimientos Técnicos.

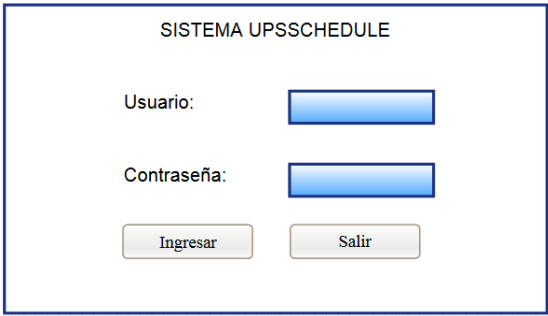
### 2.3.4. Prototipos de interfaces

Han sido diseñados, considerando criterios de simplicidad, uso intuitivo y seguridad.

#### 2.3.4.1. Ingreso al sistema

Esta pantalla constará de 1 campo de texto y un campo tipo contraseña para el ingreso de las credenciales del usuario. Además contendrá un botón para ingresar al sistema y otro para salir de la ventana de autenticación.

Prototipo



SISTEMA UPSSCHEDULE

Usuario:

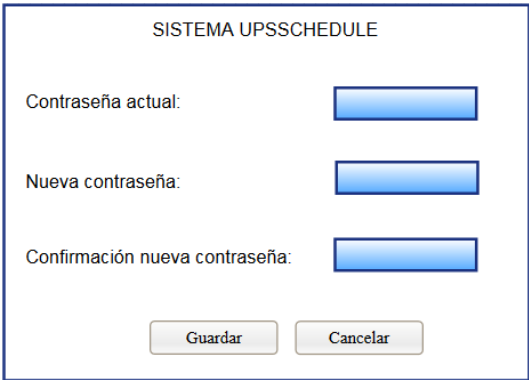
Contraseña:

*Figura 8. Prototipo Interfaz Ingreso al Sistema*  
Elaborado por: Mauricio Romero

#### 2.3.4.2. Cambio de contraseña

Esta deberá constar de 3 campos en los cuales se ingresarán la contraseña actual, la nueva contraseña y la confirmación de la nueva contraseña, además incluir botones para guardar o cancelar los cambios.

Prototipo



SISTEMA UPSSCHEDULE

Contraseña actual:

Nueva contraseña:

Confirmación nueva contraseña:

*Figura 9. Prototipo Interfaz Cambio de contraseña*  
Elaborado por: Mauricio Romero

### 2.3.4.3. Inicio

Esta pantalla contendrá 3 elementos en los cuales se podrán visualizar cuadros estadísticos con información del estado actual de la asignación de espacios.

- El primer cuadro, presentará el porcentaje de horarios con aulas asignadas en todo el campus.
- El segundo cuadro, ilustrará el porcentaje de utilización de cada bloque en la asignación existente.
- El Tercer cuadro, mostrará el porcentaje de horarios con aulas asignadas en cada carrera.

Prototipo

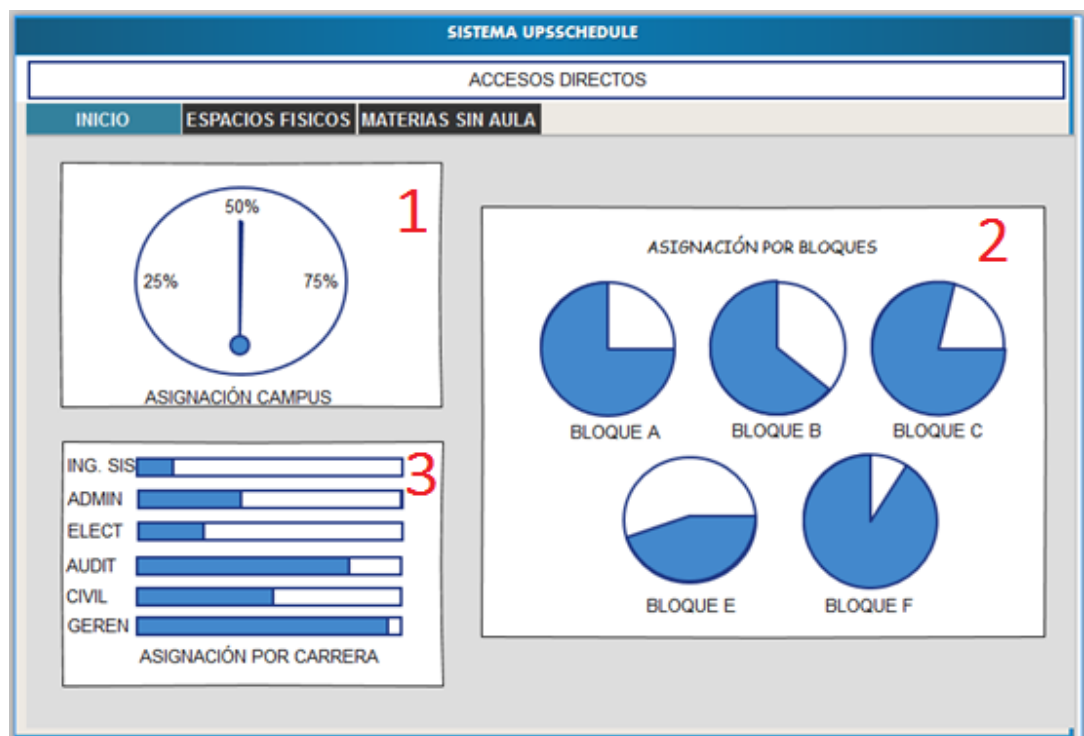


Figura 10. Prototipo Interfaz Inicio  
Elaborado por: Mauricio Romero

#### 2.3.4.4. Espacios físicos

Esta pantalla estará dividida en 3 secciones:

- La primera sección contendrá un árbol con el listado de todos los espacios físicos del campus.
- La segunda sección mostrará la leyenda de colores que identifica cada una de las carreras.
- La tercera sección presentará el horario semanal de cada aula, previamente seleccionada en el árbol de la primera sección.

## Prototipo

[illegible]

**Figura 11. Prototipo Interfaz Espacios Físicos**  
Elaborado por: Mauricio Romero

### 2.3.4.5. Materias sin aula

Esta pantalla estará divida en 3 secciones:

- La primera sección contendrá un árbol con el listado de todos los horarios que no posean un aula asignada.
- La segunda sección mostrará un listado de las aulas disponibles, que cumplan con los requerimientos de los horarios seleccionados en el árbol de la primera sección.
- La tercera sección presentará el horario semanal del aula seleccionada en la segunda sección, además la vista preliminar de la asignación de los horarios seleccionados.

## Prototipo

[illegible]

**Figura 12.** Prototipo Interfaz Materia sin Aula  
Elaborado por. Mauricio Romero

### 2.3.4.6. Reporte docente

Esta pantalla mostrará a manera de filtro un combobox con el listado de todas las carreras del Campus, incluirá además un botón para generar el reporte y otro para cancelar.

Prototipo

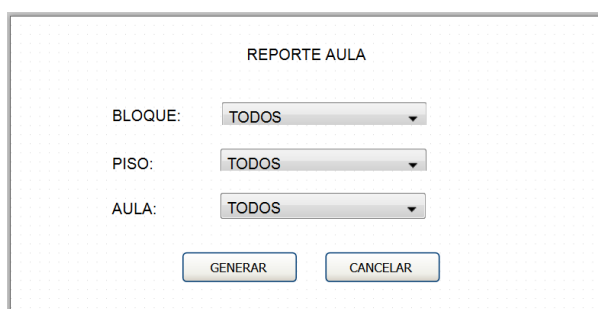
El prototipo de la interfaz 'Reporte Docente' se muestra dentro de un recuadro con un fondo de puntos. En la parte superior, el título 'REPORTE DOCENTE' está centrado. Debajo, a la izquierda, se encuentra el texto 'CARRERA:'. A la derecha de este texto hay un combobox con el valor 'TODOS' y una flecha hacia abajo. En la parte inferior del formulario, hay dos botones rectangulares: 'GENERAR' a la izquierda y 'CANCELAR' a la derecha.

*Figura 13.* Prototipo Interfaz Reporte Docente  
Elaborado por. Mauricio Romero

### 2.3.4.7. Reporte aula

Esta pantalla mostrará a manera de filtro tres combobox con los listados de todos los bloques, pisos y aulas del Campus, los que deberán mostrar la información en base a la selección jerárquica de estos (P. ej. Si se selecciona el Bloque A en el primer combobox, el segundo combobox mostrará el listado de los pisos de este bloque, de igual forma en el tercer combobox con las aulas); incluirá además, un botón para generar el reporte y otro para cancelar.

Prototipo

El prototipo de la interfaz 'Reporte Aula' se muestra dentro de un recuadro con un fondo de puntos. En la parte superior, el título 'REPORTE AULA' está centrado. Debajo, hay tres filas de filtros. La primera fila tiene el texto 'BLOQUE:' seguido de un combobox con 'TODOS'. La segunda fila tiene 'PISO:' seguido de un combobox con 'TODOS'. La tercera fila tiene 'AULA:' seguido de un combobox con 'TODOS'. En la parte inferior, hay dos botones rectangulares: 'GENERAR' a la izquierda y 'CANCELAR' a la derecha.

*Figura 14.* Prototipo Interfaz Reporte Aula  
Elaborado por. Mauricio Romero



#### 2.3.4.8. Reporte grupo

Esta pantalla mostrará a manera de filtro tres combobox con los listados de todas las carreras, niveles y grupos del Campus, los que deberán mostrar la información en base a la selección jerárquica de estos (P. ej. Si se selecciona la Carrera Ingeniería Civil en el primer combobox, el segundo combobox mostrará el listado de todos los niveles de esta Carrera, de igual forma en el tercer combobox con los grupos); incluirá además, un botón para generar el reporte y otro para cancelar.

Prototipo

El prototipo de la interfaz 'Reporte Grupo' se muestra dentro de un recuadro con un fondo de puntos. El título 'REPORTE GRUPO' está centrado en la parte superior. Debajo del título, hay tres etiquetas de texto ('CARRERA:', 'NIVEL:', 'GRUPO:') alineadas a la izquierda, cada una seguida de un menú desplegable que muestra la opción 'TODOS'. Los menús desplegables tienen una flecha hacia abajo en su extremo derecho. En la parte inferior del formulario, hay dos botones rectangulares: 'GENERAR' a la izquierda y 'CANCELAR' a la derecha.

*Figura 15.* Prototipo Interfaz Reporte Grupo  
Elaborado por: Mauricio Romero

## CAPÍTULO 3

### 3. CONSTRUCCIÓN

#### 3.1. Plataformas y herramientas

Para el desarrollo del programa Gestión de espacios físicos se considera el uso de Java como lenguaje de programación y de PostgreSQL como motor de base de datos.

Al desarrollar el programa en Java, el sistema será multiplataforma, puesto que una de las ventajas de este lenguaje, es que, puede ser ejecutado una gran variedad de Sistemas Operativos y Arquitecturas de Hardware.

Al considerar la limitación de uso de herramientas de software libre, el programa se desarrollará utilizando las siguientes herramientas:

##### 3.1.1. Netbeans IDE

Netbeans IDE es un entorno de desarrollo integrado de código abierto y gratuito, desarrollado bajo el Proyecto Netbeans fundado en Junio del 2000 por Sun Microsystems.

Este IDE es desarrollado principalmente para JAVA, y es licenciado bajo CDDL Common Development and Distribution License.

Sus principales características son:

- Multilenguaje
- Multiplataforma
- Modular y extensible

Para mayor información: <https://netbeans.org/>

### 3.1.2. JDK

Java Development Kit es un conjunto de herramientas necesario para el desarrollo de aplicaciones en Java.

El contenido principal del JDK es:

- **Herramientas de desarrollo:** desarrollar, depurar y documentar programas escritos en Java.
- **Ambiente de Ejecución Java:** posee la JVM (Java Virtual Machine) y el API de clases necesarios para la ejecución de programas Java.

Para mayor información: <http://docs.oracle.com/javase/6/docs/technotes/tools/>

### 3.1.3. Modelo Vista Controlador (MVC)

Es un patrón de diseño que consiste en separar en 3 capas diferentes el código.

- **Modelo:** esta capa se encarga del manejo de los datos.
- **Vista:** en esta capa se desarrollan todas las interfaces con las cuales interactuará el usuario.
- **Controlador:** esta capa es la encargada de realizar la comunicación entre la capa de Modelo y la capa de Vista

### 3.1.4. PostgreSQL

Es un sistema de gestión de bases de datos relacional de código abierto, se distribuye bajo licencia BSD.

Utiliza el modelo cliente/servidor e implementa el uso de multiprocesos en lugar de multihilos, garantizando la estabilidad del sistema. (Martinez)

Para mayor información: <http://www.postgresql.org/es/documentacion>

### **3.1.5. Algoritmo MD5**

MD5(Message-Digest Algorithm 5) Algoritmo de Resumen del Mensaje 5, es el algoritmo de hash mas utilizado. Utiliza 128 bits para realizar la codificación y su salida es un número hexadecimal de 32 digitos

El algoritmo MD5 tiene varias características que lo hacen particularmente interesante. En primer lugar, su aplicación a una misma entrada obtiene siempre la misma salida. Por otro lado, pequeñas diferencias en los documentos de entrada generan normalmente grandes diferencias en la salida. Además, la posibilidad de que dos documentos distintos generen la misma huella es muy baja. Para terminar, es imposible averiguar el contenido del documento original a partir de la salida. (Hacienda, f.s.)

### **3.1.6. ITextPdf**

Es una librería de código abierto que permite la creación y manipulación de documentos en formato PDF (Group).

Esta librería está disponible para Java y .NET, y soporta diferente tipos de imágenes, fuentes y fondos.

Para mayor información: <http://itextpdf.com/learn>

### **3.1.7. JCommon**

Es una librería gratuita de código abierto de propósito general utilizada por varios proyectos (JFreeChart, JFreeResport, etc.).

Se distribuye bajo licencia LGPL (Licencia Pública Reducida de GNU), actualmente su desarrollo ha sido detenido y en el futuro se eliminará la dependencia de esta librería en JFreeChart. (JFree)

### **3.1.8. JFreeChart**

Es una librería de código abierto, gratuita para el lenguaje de programación Java, permite la creación de una gran variedad de cuadros estadísticos.

Esta librería es desarrollada bajo el Proyecto JFreeChart fundada en el año 2000 y se distribuye bajo licencia LGPL.

Para mayor información: <http://www.jfree.org/jfreechart/>

### **3.1.9. Power Architect**

Es una herramienta de modelado de Base de Datos, multiplataforma con soporte a varios Sistemas de Administración como: PostgreSQL, Oracle, MS SQL Server, etc.

La versión comunitaria es open source y se distribuye gratuitamente bajo licencia GPLv3.

Para mayor información: <http://www.sqlpower.ca/page/architect>

### **3.1.10. Pencil**

Es una herramienta de código abierto, multiplataforma, diseñada para la elaboración de prototipos de Interfaces Gráficas de Usuarios (GUI), se distribuye bajo licencia GPLv2.

Para mayor información: <http://pencil.evolus.vn/Features.html>

### **3.1.11. DBVisualizer**

Es una herramienta multiplataforma para la administración de bases de datos como: Oracle, SQL Server, DB2, Sybase, MySQL, SQLite, etc. cuenta con un editor SQL para crear o modificar tablas, permite la ejecución de instrucciones SQL y la visualización de las relaciones entre las tablas en una BDD.

Para mayor información: <https://www.dbvis.com/>

### **3.1.12. Visual Paradigm para UML**

Es una herramienta para la generación de diagramas UML, posee características de ingeniería de código como generación de código, incluye también la posibilidad de generar diagramas UML a partir del código fuente mediante ingeniería inversa.

Para mayor información: <http://www.visual-paradigm.com/>

### **3.1.13. Notepad++**

Es una herramienta gratuita y de código abierto para la edición de código fuente de varios lenguajes de programación. Es escrito en C++ y se caracteriza por su velocidad de ejecución y su peso ligero. Se distribuye bajo licencia GPL y posee soporte para Windows.

Para mayor información: <http://notepad-plus-plus.org/>

### **3.1.14. Dia**

Es una herramienta gratuita de código abierto para la creación de diagramas, soporta Windows, Mac OS X y Linux. Permite el desarrollo de más de 30 tipos de diagramas y exportarlos a varios formatos.

Para mayor información: <http://dia-installer.de/>

### 3.2. Diagrama conceptual de la base de datos UPSSchedule

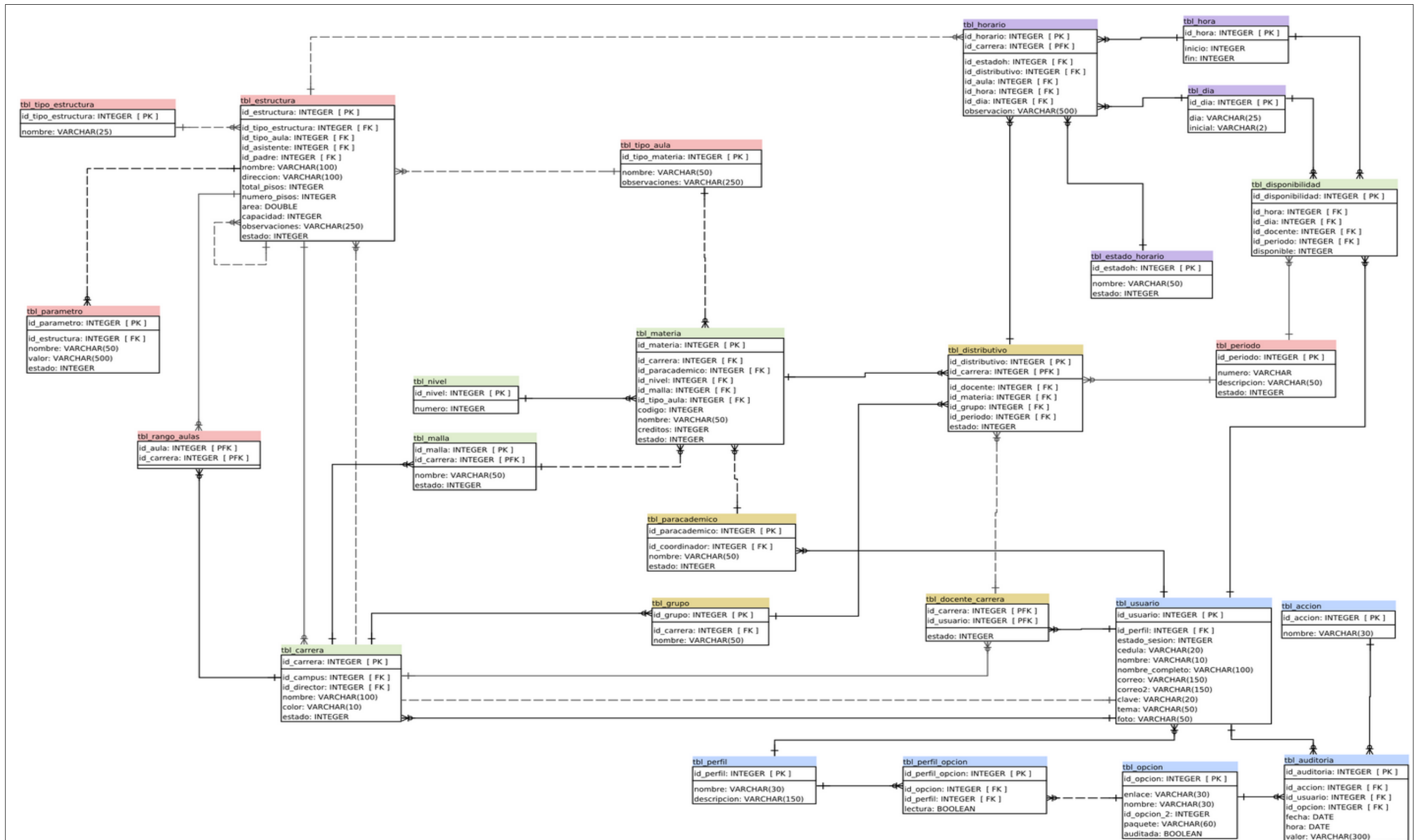


Figura 16. Diagrama Conceptual BDD UPSSchedule

Elaborado por: Mauricio Romero

Se observan las tablas contenidas en la BDD UPSSchedule y las relaciones necesarias para el correcto funcionamiento de los diferentes módulos que integran el Proyecto UPS Schedule.

### 3.2.1. Diccionario de datos

A continuación se presentará el diccionario de datos con las tablas que son utilizadas por el Programa “Gestión de Espacios Físicos”.

#### 3.2.1.1. Tbl\_distributivo

Almacena la información relaciona entre materia, grupo y docentes.

Tabla 15. Descripción de la tabla *tbl\_distributivo*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_distributivo	serial		X		X	Campo que identifica la tabla.
id_carrera	integer				X	Campo que identifica la carrera a la cual pertenece.
id_docente	integer					Campo que identifica el docente al cual pertenece el distributivo.
id_materia	integer			X	X	Campo que identifica la materia a la cual pertenece el distributivo.
id_grupo	integer			X	X	Campo que identifica el grupo al cual pertenece el distributivo.
id_periodo	integer			X	X	Campo que identifica el periodo al cual pertenece el distributivo.
estado	integer				X	Campo que identifica si el registro debe considerarse o no.

Nota.

Elaborado por: Mauricio Romero

#### 3.2.1.2. Tbl\_carrera

Almacena la información de las carreras.

Tabla 16. Descripción de la tabla *tbl\_carrera*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_carrera	serial		X		X	Campo que identifica la tabla
id_campus	integer			X	X	Campo que identifica el campus al cual pertenece la materia
id_director	integer			X	X	Campo que identifica el director de la carrera
nombre	character varying	100			X	Campo que contiene el nombre de la materia
color	character varying	10			X	Campo que contiene el color de la materia
estado	integer				X	Campo que identifica si el registro debe considerarse o no

Nota.

Elaborado por: Mauricio Romero



### 3.2.1.3. Tbl\_dia

Almacena los días de la semana en los que, se dictan clases.

Tabla 17. Descripción de la tabla *tbl\_dia*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_dia	serial		X		X	Campo que identifica la tabla
dia	character varying	25			X	Campo que contiene el nombre del día
inicial	character varying	2			X	campo que contiene las iniciales del día

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.4. Tbl\_estructura

Almacena la estructura física de la universidad: sedes, campus, etc.

Tabla 18. Descripción de la tabla *tbl\_estructura*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_estructura	serial		X		X	Campo que identifica la tabla
id_tipo_estructura	integer			X		Campo que identifica el tipo al cual pertenece la estructura
id_tipo_aula	integer			X		Campo que identifica el tipo de aula cuando la estructura es un aula
id_asistente	integer			X		Campo que identifica el administrador de la estructura
id_padre	integer			X		Campo que identifica la estructura a la cual pertenece
nombre	character varying	100			X	Campo que contiene el nombre de la estructura
direccion	character varying	100				Campo que contiene la dirección de la estructura
total_pisos	integer				X	Campo que contiene el número de pisos de la estructura
numero_pisos	integer				X	Campo que contiene el número de piso en el cual está la estructura
area	double precision				X	Campo que contiene el área de la estructura
capacidad	integer				X	Campo que contiene la capacidad de la estructura
observaciones	character varying	250			X	Campo que contiene observaciones sobre la estructura
estado	integer				X	Campo que identifica si el registro debe considerarse o no

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.5. Tbl\_estado\_horario

Almacena los estados que puede tener un horario: cruce, correcto, etc.

Tabla 19. Descripción de la tabla *tbl\_estado\_horario*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_estadoh	serial		X		X	Campo que identifica la tabla
nombre	character varying	50			X	Campo que contiene el nombre del tipo de estado que puede tener un horario
estado	integer				X	Campo que identifica si el registro debe considerarse o no
color	character varying	10				Campo que contiene el color del estado que puede obtener un estado

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.6. Tbl\_grupo

Almacena la información de los grupos (paralelos) que dispone cada carrera.

Tabla 20. Descripción de la tabla *tbl\_grupo*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_grupo	serial		X		X	Campo que identifica la tabla
id_carrera	integer			X	X	Campo que identifica la carrera a la cual pertenece el grupo
nombre	character varying	50			X	Campo que contiene el nombre del grupo

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.7. Tbl\_hora

Almacena las horas en la cuales se imparten clases.

Tabla 21. Descripción de la tabla *tbl\_hora*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_hora	serial		X		X	Campo que identifica la tabla
inicio	integer				X	Campo que contiene el inicio de la hora
fin	integer				X	Campo que contiene el fin de la hora

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.8. Tbl\_malla

Almacena la información de las mallas académicas vigentes.

Tabla 22. Descripción de la tabla *tbl\_malla*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_malla	serial		X		X	Campo que identifica la tabla
id_carrera	integer			X	X	Campo que identifica la carrera a la cual pertenece la malla
nombre	character varying				X	Campo que contiene el nombre de la malla
estado	integer				X	Campo que identifica si el registro debe considerarse o no

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.9. Tbl\_nivel

Almacena la información de los niveles.

Tabla 23. Descripción de la tabla *tbl\_nivel*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_nivel	serial		X		X	Campo que identifica la tabla
numero	integer				X	Campo que contiene el número del nivel
nombre	character varying	20				Campo que contiene el nombre del nivel

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.10. Tbl\_horario

Almacena la información de horarios por hora y día de la semana.

Tabla 24. Descripción de la tabla *tbl\_horario*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_horario	serial		X		X	Campo que identifica la tabla
id_carrera	integer			X	X	Campo que identifica la carrera a la cual pertenece el horario
id_estadoh	integer			X	X	Campo que contiene el estado del horario
id_distributivo	integer			X	X	Campo que identifica el distributivo al cual pertenece el horario
id_aula	integer			X		Campo que identifica el aula al cual pertenece el horario
id_hora	integer			X	X	Campo que identifica la hora del horario
id_dia	integer			X	X	Campo que identifica el día del horario
observacion	character varying	500			X	Campo que contiene observaciones sobre el horario

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.11. Tbl\_mencion

Almacena la información de las menciones.

Tabla 25. Descripción de la tabla *tbl\_mencion*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_mencion	integer		X		X	Campo que identifica la tabla
nombre	character varying	50			X	Campo que contiene el nombre de la mención

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.12. Tbl\_materia

Almacena la información de las materias.

Tabla 26. Descripción de la tabla *tbl\_materia*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_materia	serial		X		X	Campo que identifica la tabla
id_carrera	integer			X	X	Campo que identifica la carrera a la cual pertenece la materia
id_paracademico	integer			X		Campo que identifica a que paracademico pertenece la materia
id_nivel	integer			X	X	Campo que identifica a que nivel pertenece la materia
id_malla	integer			X	X	Campo que identifica a que malla pertenece la materia
id_tipo_aula	integer			X	X	Campo que identifica que tipo de aula requiere la materia
codigo	integer				X	Campo que contiene el código SNA de la materia
nombre	character varying	50			X	Campo que contiene el nombre de la materia
creditos	integer				X	Campo que contiene el número de créditos de la materia
estado	integer				X	Campo que identifica si el registro debe considerarse o no
id_mencion	integer			X		Campo que identifica la mención a la cual pertenece la materia

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.13. Tbl\_rango\_aulas

Almacena registros de aulas en las cuales se debe realizar la asignación de carrera.

Tabla 27. Descripción de la tabla tbl\_rango\_aulas

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_aula	integer			X	X	Campo que identifica el aula
id_carrera	integer			X	X	Campo que identifica la carrera

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.14. Tbl\_periodo

Almacena la información de los periodos.

Tabla 28. Descripción de la tabla tbl\_periodo

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_periodo	integer		X		X	Campo que identifica la tabla
numero	character varying				X	Campo que contiene el número del periodo
descripcion	character varying	50			X	Campo que contiene la descripción del periodo
estado	integer				X	Campo que identifica si el registro debe considerarse o no

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.15. Tbl\_tipo\_estructura

Almacena los tipos de estructura: sede, campus, etc.

Tabla 29. Descripción de la tabla tbl\_tipo\_estructura

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_tipo_estructura	serial		X		X	Campo que identifica la tabla
nombre	character varying	25			X	Campo que contiene el nombre del tipo de estructura

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.16. Tbl\_tipo\_aula

Almacena los tipos de aula: regular, laboratorio, etc.

Tabla 30. Descripción de la tabla tbl\_tipo\_aula

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_tipo_materia	serial		X		X	Campo que identifica la tabla
nombre	character varying	50			X	Campo que contiene el nombre del tipo de aula
observaciones	character varying	250			X	Observaciones del tipo de aula

Nota.

Elaborado por: Mauricio Romero

### 3.2.1.17. Tbl\_usuario

Almacena la información de los diferentes tipos de usuarios.

Tabla 31. Descripción de la tabla *tbl\_usuario*

Atributo	Tipo	Tam	PK	FK	Not Null	Descripción
id_usuario	Serial		X		X	Campo que identifica la tabla
id_perfil	integer				X	Campo que identifica el perfil de usuario
estado_sesion	integer				X	Campo que identifica si esta autenticado el usuario
cedula	character varying	20			X	Campo que contiene la cedula del usuario
nombre	character varying	10			X	Campo que contiene el nombre del usuario
nombre_completo	character varying	100			X	Campo que contiene el nombre completo del usuario
correo	character varying	150				Campo que contiene el correo del usuario
correo2	character varying	150				Campo que contiene un segundo correo del usuario
clave	character varying	200			X	Campo que contiene la clave del usuario
tema	character varying	50			X	Campo que identifica el tema del usuario
foto	character varying	50				Campo que contiene la foto del usuario

Nota.

Elaborado por: Mauricio Romero

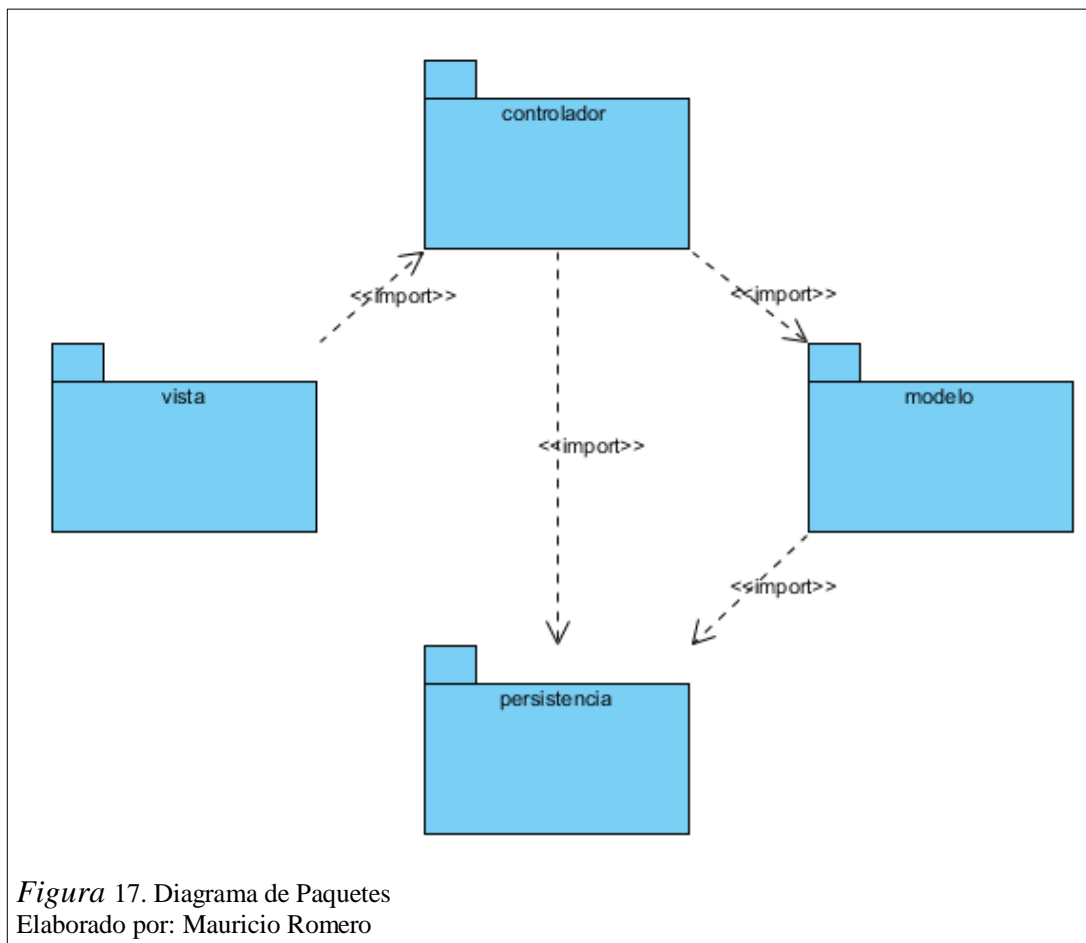
### 3.3. Estructura de paquetes y clases

El programa “Gestión de Espacios Físicos” ha sido desarrollado utilizando el lenguaje de programación Java, utilizando el patrón de diseño MVC (Modelo – Vista – Controlador).

Las clases han sido separadas en los siguientes paquetes en base a su función:

- Persistencia
- Modelo
- Vista
- Controlador

### 3.3.1. Diagrama de paquetes



Este diagrama muestra los paquetes de clases que conforman el Sistema Gestión de Espacios Físicos, y la relación que tienen.

El paquete vista se comunica con el paquete controlador y este a su vez se comunica con los paquetes persistencia y modelo, de igual forma el paquete modelo se comunica con el paquete persistencia. A continuación se presentará la explicación de cada paquete.

### 3.3.2. Paquete Vista

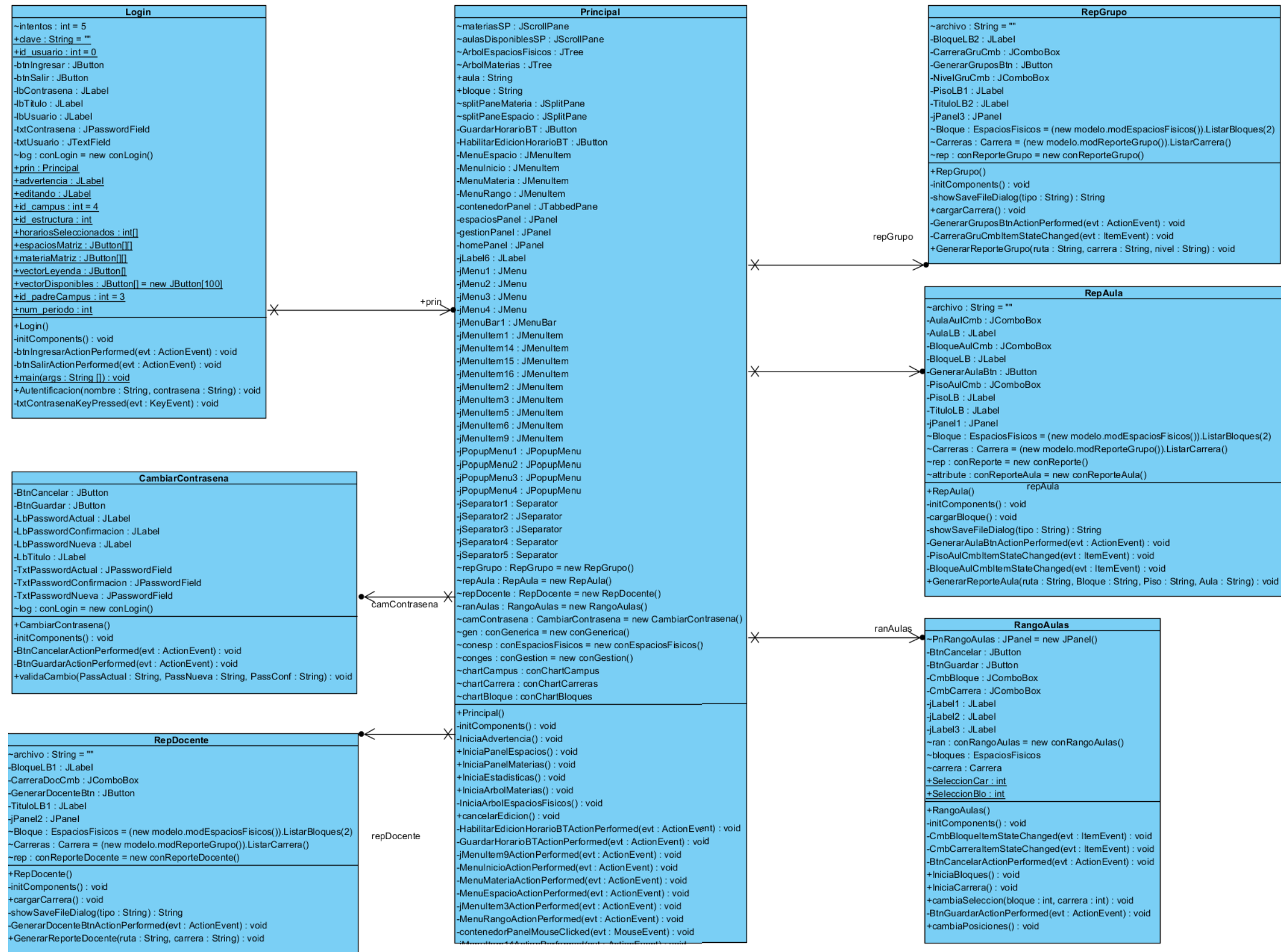


Figura 18. Diagrama de Clases del Paquete Vista  
Elaborado por: Mauricio Romero



Este diagrama muestra las clases y las relaciones existentes dentro del paquete vista, este paquete contiene las interfaces del usuario, es el encargado de la comunicación entre el usuario y el sistema, consta de 6 clases de tipo JFrame, que se describen a continuación.

### 3.3.2.1. Clase Login

Es la primera interfaz del programa y se encarga de presentar campos de texto en los cuales el usuario deberá ingresar sus credenciales de autenticación.

#### Método Autenticacion

```
public void Autenticacion(String nombre, String contrasena) {
    if (!clave.contains("")) {
        if (log.VerificaUsuario(nombre, contrasena)) {
            this.dispose();
            prin = new Principal();
            prin.setVisible(true);
        } else {
            intentos--;
            JOptionPane.showMessageDialog(null, "Usuario o contraseña incorrecta.
" + "Intentos restantes " + intentos, "Error de login", JOptionPane.ERROR_MESSAGE);
            if (intentos == 0) {
                System.exit(0);
            }
        }
    } else {
        JOptionPane.showMessageDialog(null, "Por favor no intente utilizar SQL
Injection", "Error de login", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
}
```

Este método se encarga de enviar las credenciales proporcionadas por el usuario, al paquete controlador para realizar la validación de estas, en el caso de que sean correctas se mostrará la interfaz Principal, caso contrario se mostrará un mensaje de error.

El proceso de autenticación permite hasta 4 intentos fallidos, posterior a estos, se cerrará el sistema.

### 3.3.2.2. Clase Principal

Esta interfaz, contiene o brinda acceso a todas las funcionales del programa.

Muestra pestañas para cumplir las funcionalidades de: visualización de estadísticas, asignación de materias sin aula y gestión de espacios físicos.

Contiene accesos a las siguientes interfaces:

- Rango de Aulas
- Cambiar contraseña
- Reportes

La información contenida en esta interfaz se basa en el campus al cual está relacionado el usuario autenticado.

### Método IniciaEstadisticas

```
public void IniciaEstadisticas() {
    homePanel.removeAll();
    chartCampus = new conChartCampus();
    chartCarrera = new conChartCarreras();
    chartBloque = new conChartBloques();
    chartCampus.setBounds(130, 45, 300, 260);
    chartCarrera.setBounds(30, 310, 500, 300);
    chartBloque.setBounds(550, 15, 800, 580);
    JLabel jLabel1 = new JLabel();
    jLabel1.setFont(new java.awt.Font("SansSerif", 1, 20));
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("ASIGNACIÓN DEL CAMPUS");
    jLabel1.setBounds(110, 15, 300, 20);
    homePanel.add(jLabel1);
    homePanel.add(chartCarrera);
    homePanel.add(chartCampus);
    homePanel.add(chartBloque);
}
```

Este método se encarga de inicializar los objetos necesarios para visualizar los cuadros estadísticos que muestran el estado de la asignación por campus, bloques y carreras.

### Método IniciaArbolMaterias

```
public void IniciaArbolMaterias() {
    ArbolMaterias = conges.GenerarArbolHorarios();
    ArbolMaterias.addTreeSelectionListener(new TreeSelectionListener() {
        @Override
        public void valueChanged(TreeSelectionEvent e) {
            TreePath[] paths = ArbolMaterias.getSelectionPaths();
            conges.AulasDisponiblesLimpia();
            conges.LimpiarHorario();
            if (paths != null) {
                boolean horario = true;
                for (int i = 0; i < paths.length; i++) {
                    if (paths[i].getPathCount() != 6) {
                        horario = false;
                    }
                }
            }
            if (horario) {
```

```

        conges.AulasDisponibles(paths);
    }
}
});
materiasSP = new JScrollPane(ArbolMaterias);
materiasSP.setFont(gen.fuente);
materiasSP.setBounds(30, 30, 230, 280);
materiasSP.setBorder(new LineBorder(Color.gray, 1));
materiasSP.setBorder(null);
splitPaneMateria = new JSplitPane(JSplitPane.VERTICAL_SPLIT,
    materiasSP, aulasDisponiblesSP);
splitPaneMateria.setOneTouchExpandable(true);
splitPaneMateria.setDividerLocation(350);
splitPaneMateria.setBounds(30, 45, 270, 560);
gestionPanel.add(splitPaneMateria);
}

```

Este método se encarga de inicializar el árbol materia de la pestaña Materias sin Aula, que contiene un listado de los horarios a los cuales no se asignaron aulas, de las materias pertenecientes al campus relacionado al usuario autenticado.

## Método IniciaArbolEspaciosFísicos

```

private void IniciaArbolEspaciosFisicos() {
    ArbolEspaciosFisicos = conesp.GenerarArbolEspacios();
    ArbolEspaciosFisicos.addTreeSelectionListener(new TreeSelectionListener() {
        @Override
        public void valueChanged(TreeSelectionEvent e) {
            TreePath[] paths = ArbolEspaciosFisicos.getSelectionPaths();
            conesp.LimpiarHorario();
            if (paths != null) {
                if (paths[0].getPathCount() == 4 && paths.length == 1) {
                    aula = paths[0].getLastPathComponent().toString();
                    bloque = paths[0].getPathComponent(1).toString();
                    conesp.BuscarHorario(aula, bloque);
                    if
(HabilitarEdicionHorarioBT.getToolTipText().equals("EDITANDO")) {
                        cancelarEdicion();
                    }
                }
            }
        }
    });
    JScrollPane espacios = new JScrollPane(ArbolEspaciosFisicos);
    espacios.setBorder(null);
    espacios.setBounds(20, 40, 290, 360);
    splitPaneEspacio = new JSplitPane(JSplitPane.VERTICAL_SPLIT,
        espacios, iniciaVectorLeyenda());
    splitPaneEspacio.setOneTouchExpandable(true);
    splitPaneEspacio.setDividerLocation(350);
    splitPaneEspacio.setBounds(30, 45, 270, 560);
    espaciosPanel.add(splitPaneEspacio);
}

```

Este método se encarga de inicializar el árbol Espacios Físicos de la pestaña Espacios Físicos, que contiene un listado de los espacios físicos relacionados al campus al que pertenece el usuario autenticado.

A su vez se encarga de especificar las tareas a realizar cuando se seleccione un elemento del árbol.

### 3.3.2.3. Clase CambiarContrasena

Esta interfaz permite al usuario cambiar la contraseña con la cual se autentifica en el sistema, para realizar esta operación el usuario debe ingresar su contraseña actual, la nueva contraseña y la confirmación de la nueva contraseña.

#### Método CambiarContrasena

```
public void validaCambio(String PassActual, String PassNueva, String PassConf) {
    if (PassActual.isEmpty() || PassConf.isEmpty() || PassNueva.isEmpty()) {
        JOptionPane.showMessageDialog(null, "Todos los campos son requeridos",
            "Error al intentar cambiar la contraseña", JOptionPane.ERROR_MESSAGE);
    } else if (!PassNueva.equals(PassNueva)) {
        JOptionPane.showMessageDialog(null, "Las contraseñas no coinciden",
            "Error al intentar cambiar la contraseña", JOptionPane.ERROR_MESSAGE);
    } else if (PassActual.equals(clave)) {
        log.CambiarContrasena(PassNueva);
        JOptionPane.showMessageDialog(null, "Éxito al cambiar la contraseña",
            "Éxito", JOptionPane.INFORMATION_MESSAGE);
    }
}
```

Este método, valida el formato de las credenciales, las envía a los métodos encargados de realizar el cambio y muestra mensajes informativos en el caso de éxito o error en el proceso.

### 3.3.2.4. Clase RangoAulas

Esta interfaz permite gestionar en forma amigable e intuitiva los grupos de aulas preasignadas que posee cada carrera, estos grupos de aulas son utilizados por el módulo de asignación automática.

#### Método CambiaSeleccion

```
public void cambiaSeleccion(int bloque, int carrera) {
    PnRangoAulas.setVisible(false);
    PnRangoAulas = ran.RangoAulas(bloque, carrera);
    PnRangoAulas.setBounds(30, 150, 40 + 65 * maxAulas, 31 * maxPisos);
    this.add(PnRangoAulas);
    cambiaPosiciones();
}
```

Este método envía los nombres de carrera y el bloque seleccionados en la interfaz al paquete Controlador y actualiza el panel encargado de mostrar los grupos de aulas preasignadas a cada carrera.

### 3.3.2.5. Clase RepAula

Esta Interfaz permite al usuario generar un reporte con los horarios que se dictan en cada una de las aulas, muestra opciones de filtro por Bloque, Piso y aula.

Presionar el botón Guardar de esta interfaz, despliega una ventana que permite al usuario dar ruta y nombre al reporte a generar.

#### Método GenerarReporteAula

```
public void GenerarReporteAula(String ruta, String Bloque, String Piso, String Aula)
{
    try {
        if (!ruta.equalsIgnoreCase("")) {
            archivo = ruta + "_tmp.pdf";
            rep.generaReporteAula(Bloque, Piso, Aula, archivo, ruta + ".pdf");
            File file = new File(archivo);
            file.delete();
            Desktop d = java.awt.Desktop.getDesktop();
            this.dispose();
            prin.dispose();
            JOptionPane.showMessageDialog(null, "Reporte generado con exito",
"Éxito", JOptionPane.INFORMATION_MESSAGE);
            d.open(new java.io.File(ruta + ".pdf"));
            prin.setVisible(true);
        }
    } catch (DocumentException | HeadlessException | IOException e) {
        System.out.println("error");
        JOptionPane.showMessageDialog(null, "Error " + e.getMessage(), "Error al
generar el reporte", JOptionPane.ERROR_MESSAGE);
    }
}
```

Este método recibe la ruta en la que se generará el archivo y los filtros por bloque, piso y aula, envía estos valores al paquete Controlador, informa al usuario en caso de éxito o error y despliega el archivo generado.

### 3.3.2.6. Clase RepDocente

Esta Interfaz permite al usuario generar un reporte con los horarios y aulas que posee cada Docente, muestra opciones de filtro por Carrera.

Presionar el botón Guardar de esta interfaz, despliega una ventana que permite al usuario dar ruta y nombre al reporte a generar.

#### Método GenerarReporteDocente

```
public void GenerarReporteDocente(String ruta, String carrera) {
    try {
        if (!ruta.equalsIgnoreCase("")) {
            archivo = ruta + "_tmp.pdf";
            rep.generaReporteDocente(carrera, archivo, ruta + ".pdf");
            File file = new File(archivo);
            file.delete();
            Desktop d = java.awt.Desktop.getDesktop();
            JOptionPane.showMessageDialog(null, "Reporte generado con exito",
"Éxito", JOptionPane.INFORMATION_MESSAGE);
            d.open(new java.io.File(ruta + ".pdf"));
        }
    }
}
```

```

        } catch (DocumentException | HeadlessException | IOException e) {
            JOptionPane.showMessageDialog(null, "Error " + e.getMessage(), "Error al
generar el reporte", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

Este método recibe la ruta en la que se generará el archivo y el filtro por carrera, envía estos valores al paquete Controlador, informa al usuario en caso de éxito o error y despliega el archivo generado.

### 3.3.2.7. Clase RepGrupo

Esta Interfaz permite al usuario generar un reporte con las aulas asignadas a cada grupo, muestra opciones de filtro por Carrera, Nivel y Grupo

Presionar el botón Guardar de esta interfaz, despliega una ventana que permite al usuario dar ruta y nombre al reporte a generar.

#### Método GeneraReporteGrupo

```

public void GeneraReporteGrupo(String ruta, String carrera, String nivel) {
    try {
        if (!ruta.equalsIgnoreCase("")) {
            archivo = ruta + "_tmp.pdf";
            rep.generaReporteGrupo(carrera, nivel, archivo, ruta + ".pdf");
            File file = new File(archivo);
            file.delete();
            Desktop d = java.awt.Desktop.getDesktop();
            JOptionPane.showMessageDialog(null, "Reporte generado con exito",
"Éxito", JOptionPane.INFORMATION_MESSAGE);
            d.open(new java.io.File(ruta + ".pdf"));
        }
    } catch (DocumentException | HeadlessException | IOException e) {
        JOptionPane.showMessageDialog(null, "Error " + e.getMessage(), "Error al
generar el reporte", JOptionPane.ERROR_MESSAGE);
    }
}

```

Este método recibe la ruta en la que se generará el archivo y los filtros por carrera y nivel, envía estos valores al paquete Controlador, informa al usuario en caso de éxito o error y despliega el archivo generado.

3.3.3. Paquete Controlador

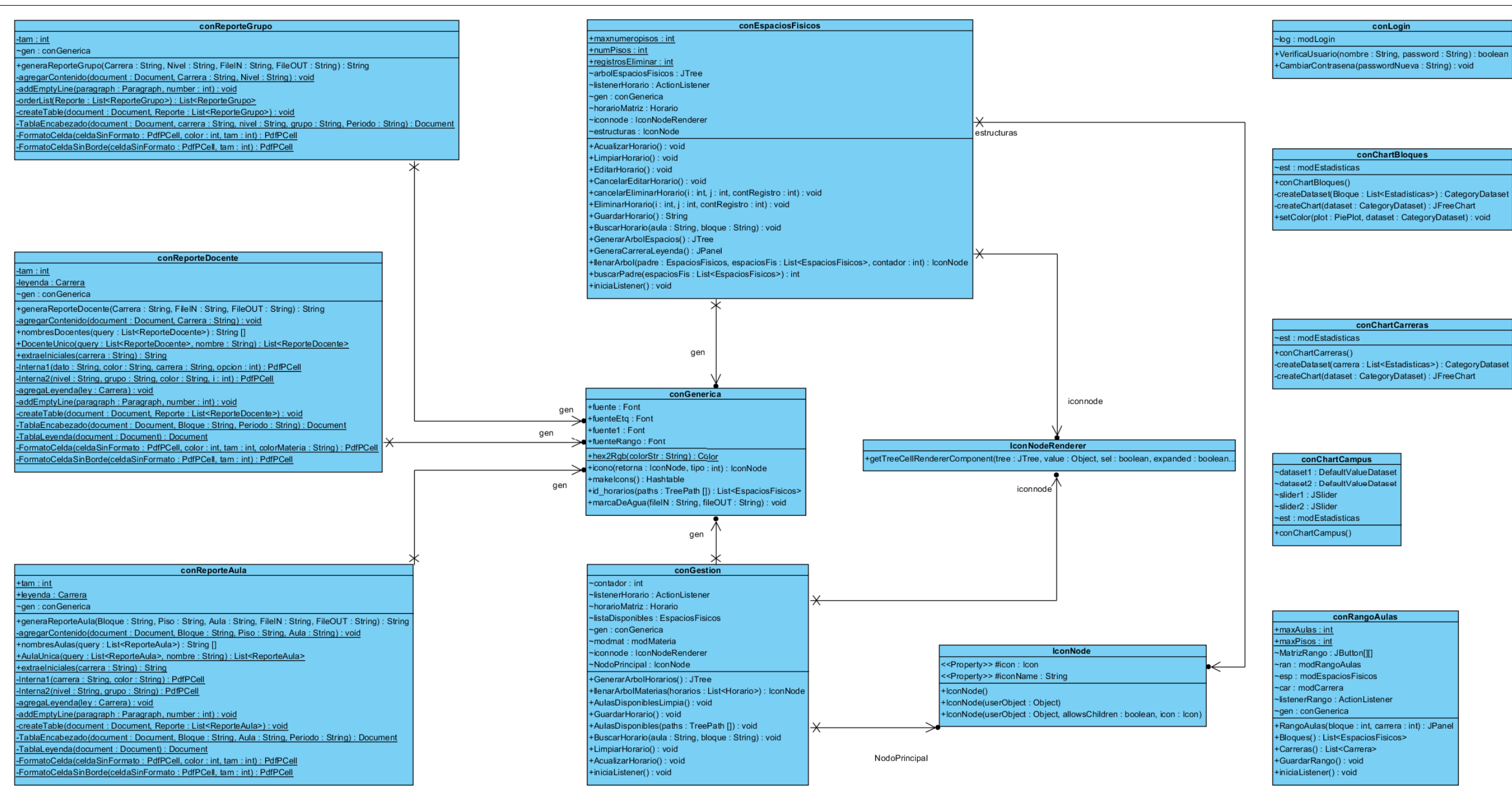


Figura 19. Diagrama de clases del Paquete Controlador  
Elaborado por: Mauricio Romero

Este diagrama muestra las clases y las relaciones existentes dentro del paquete controlador, este paquete es el encargado de responder a las peticiones del usuario, recibe las peticiones recibidas en las interfaces y solicita al paquete modelo la información necesaria para responder a dichas peticiones, consta de 13 clases que se describen a continuación.

#### **3.3.3.1. Clase ConLogin**

Se encarga de enviar las credenciales del usuario a la clase modLogin y de informar a la interfaz Principal los resultados de la verificación.

#### **Métodos VerificaUsuario - CambiarContraseña**

```
public boolean VerificaUsuario(String nombre, String password)
{
    return log.VerificaLogin(nombre, password);
}

public void CambiarContraseña(String passwordNueva)
{
    log.CambiarContraseña(passwordNueva);
}
```

Estos métodos son los encargados de direccionar las solicitudes de autenticación y cambio de contraseña al paquete modLogin.

#### **3.3.3.2. Clase ConChartCampus**

Esta clase es la encargada de generar el panel y el cuadro estadístico sobre el estado de la asignación de aulas a nivel del campus, utiliza elementos de la librería JFreeChart.

Recibe la solicitud de la clase Principal, y solicita información a la clase modEstadísticas.

#### **3.3.3.3. Clase ConChartCarrera**

Esta clase es la encargada de generar el panel y el cuadro estadístico sobre el estado de la asignación de aulas por carreras, utiliza elementos de la librería JFreeChart.

Recibe la solicitud de la clase Principal, y solicita información a la clase modEstadísticas.



## Método CreateDataSet

```
private CategoryDataset createDataset(List<persistencia.Estadisticas> carrera) {
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();

    double enUso;
    double total;
    double promedio;
    for (int i = 0; i < carrera.size(); i++) {
        enUso = (double) carrera.get(i).enUso;
        total = (double) carrera.get(i).total;
        promedio = enUso / total;
        dataset.setValue(promedio, "ASIGNADO", carrera.get(i).Nombre);
        dataset.setValue(1 - promedio, "SIN ASIGNAR", carrera.get(i).Nombre);
    }
    return dataset;
}
```

Este método es el encargado de generar el grupo de valores que contendrá el cuadro estadístico de la asignación realizada a cada carrera, ejecuta un loop para recorrer los registros enviados por el modelo.

### 3.3.3.4. Clase ConChartBloques

Esta clase es la encargada de generar el panel y el cuadro estadístico sobre el estado de la asignación de aulas por bloques, utiliza elementos de la librería JFreeChart.

Recibe la solicitud de la clase Principal, y solicita información a la clase modEstadisticas.

## Método CreateDataset

```
private CategoryDataset createDataset(List<persistencia.Estadisticas> Bloque) {
    final DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    double promedio;
    for (Estadisticas Bloque1 : Bloque) {
        promedio = (Bloque1.enUso * 100) / Bloque1.total;
        dataset.addValue(promedio, Bloque1.Nombre, "ASIGNADA");
        dataset.addValue(100 - promedio, Bloque1.Nombre, "SIN ASIGNAR");
    }
    return dataset;
}
```

Este método es el encargado de generar el grupo de valores que contendrá el cuadro estadístico de la asignación realizada en cada bloque, ejecuta un loop para recorrer los registros enviados por el modelo.

### 3.3.3.5. Clase ConRangoAulas

Esta clase es la encargada de generar la grilla de botones que se despliega en la Interfaz RangoAulas, de igual manera se encarga de asignar las tareas a realizar cuando el usuario presione algún elemento de la grilla.

## Método GuardarRango

```
public void GuardarRango() {
    for (int i = 0; i < maxPisos; i++) {
        for (int j = 0; j < maxAulas; j++) {
            if (MatrizRango[i][j] != null) {
                String[] pisoAula = MatrizRango[i][j].getActionCommand().split("
");

                int est = Integer.parseInt(pisoAula[0]);
                int carr = Integer.parseInt(pisoAula[3]);
                if (MatrizRango[i][j].getBackground() == Color.RED) {
                    ran.EliminaRango(est, carr);
                } else if (MatrizRango[i][j].getBackground() == Color.GREEN) {
                    ran.InsertaRango(est, carr);
                }
            }
        }
    }
}
```

Este método se encarga de analizar los elementos de la grilla de rangos de Aulas, determina si se asignará o se eliminará la relación entre un espacio físico y una carrera, y envía los valores de las estructuras y carreras al paquete modelo para realizar las operaciones requeridas.

### 3.3.3.6. Clase ConEspaciosFisicos

Esta clase se encarga de llenar el árbol y la grilla de botones de la pestaña Espacios Físicos en la Interfaz Principal, se relaciona con el modelo para obtener el listado de todos los espacios físicos relacionados al usuario autenticado.

Se relaciona con las clases ICONNODE e ICONNODERENDER para modificar los íconos por defecto que utiliza el árbol de espacios físicos.

## Método ActualizarHorario

```
public void ActualizarHorario() {
    LimpiarHorario();
    if (horarioMatriz != null) {
        if (!horarioMatriz.isEmpty()) {
            LimpiarHorario();
            int contRegistro = 0;
            int Dia = horarioMatriz.get(contRegistro).id_dia;
            int Hora = horarioMatriz.get(contRegistro).id_hora;
            Font fuente = new Font("TimesNewRoman", Font.BOLD, 7);
            for (int i = 0; i < 7; i++) {
                for (int j = 0; j < 14; j++) {
                    if (j == Hora - 1 && i == Dia - 1) {
                        espaciosMatriz[i][j].setFont(fuente);
                        espaciosMatriz[i][j].setText("<html><center>" +
horarioMatriz.get(contRegistro).materia + "<br>"
+ horarioMatriz.get(contRegistro).grupo + "<br>"
+ horarioMatriz.get(contRegistro).nivel +
"</center></html>");
                        espaciosMatriz[i][j].setBackground(gen.hex2Rgb(horarioMatriz.get(contRegistro).color)
);
                        espaciosMatriz[i][j].setToolTipText("<html>" +
horarioMatriz.get(contRegistro).carrera + "<br>"
+ horarioMatriz.get(contRegistro).docente +
"</html>");
                    }
                }
            }
        }
    }
}
```

```
contRegistro);  
    espaciosMatriz[i][j].setActionCommand(i + " " + j + " " +  
    espaciosMatriz[i][j].setEnabled(true);  
    contRegistro++;  
    if (contRegistro < horarioMatriz.size()) {  
        Dia = horarioMatriz.get(contRegistro).id_dia;  
        Hora = horarioMatriz.get(contRegistro).id_hora;  
    }  
}  
}  
}  
}
```

Se encarga de actualizar la grilla de botones que muestra el horario semanal de un aula seleccionada en la pestaña Espacios Físicos de la Interfaz Principal.

### 3.3.3.7. Clase ConGestion

Llena el árbol de materias sin aula, la grilla de horarios y el panel de aulas disponibles de la pestaña Materia sin Aulas de la Interfaz Principal, se comunica con las clases `ICONNODE` e `ICONNODERENDER` para utilizar íconos personalizados en el árbol de materias sin aula.

## Método AulasDisponibles

```
public void AulasDisponibles(TreePath[] paths) {
    inicialListener();
    horariosSeleccionados = new int[paths.length];
    listaDisponibles = gen.id_horarios(paths);
    if (!listaDisponibles.isEmpty()) {
        for (int i = 0; i < listaDisponibles.size(); i++) {
            vectorDisponibles[i].setBackground(new Color(27, 161, 226));
            LimpiarHorario();
            vectorDisponibles[i].setText(listaDisponibles.get(i).nombre);
            vectorDisponibles[i].setToolTipText(listaDisponibles.get(i).bloque);
            vectorDisponibles[i].setVisible(true);
            vectorDisponibles[i].addActionListener(listenerHorario);
            vectorDisponibles[i].setActionCommand(i + "");
        }
    }
}
```

Se encarga de mostrar las aulas disponibles que cumplen con los requerimientos de hora, día y tipo de aula, de los horarios seleccionados en el árbol materias sin aula.

### 3.3.3.8. Clase ConReporteAula

Esta clase es la encargada de generar el reporte de la asignación por aula, recibe filtros por Bloque, Piso y Aula de la Interfaz RepAula.

## Método GenerarReporteAula

```
public String generaReporteAula (String Bloque, String Piso, String Aula, String
FileIN, String FileOUT) throws DocumentException, FileNotFoundException
{
    try
    {
        Document reporte = new Document();
        PdfWriter.getInstance(reporte, new FileOutputStream(FileIN));
        reporte.setPageSize(PageSize.A4.rotate());
        reporte.setMargins(36,36,36,36);
        reporte.setMarginMirroringTopBottom(true);
        reporte.open();
        agregarContenido(reporte, Bloque, Piso, Aula);
        reporte.close();
        gen.marcaDeAgua (FileIN,FileOUT);
        return "exito";
    }
    catch (Exception e)
    {
        return e.getMessage();
    }
}
```

Este método es el encargado de generar el archivo PDF con la información de la Asignación por cada Aula. Se encarga de crear el archivo, llamar a los métodos que realizan la carga de información y cerrar el archivo.

### 3.3.3.9. Clase ConReporteDocente

Esta clase es la encargada de generar el reporte de la asignación por Docente, recibe filtro por Carrera de la Interfaz RepDocente.

## Método GenerarReporteDocente

```
public String generaReporteDocente (String Carrera, String FileIN, String FileOUT)
throws DocumentException, FileNotFoundException
{
    try
    {
        Document reporte = new Document();
        PdfWriter.getInstance(reporte, new FileOutputStream(FileIN));
        reporte.setPageSize(PageSize.A4.rotate());
        reporte.setMargins(36,36,36,36);
        reporte.setMarginMirroringTopBottom(true);
        reporte.open();
        agregarContenido(reporte, Carrera);
        reporte.close();
        gen.marcaDeAgua (FileIN,FileOUT);
        return "exito";
    }
    catch (FileNotFoundException | DocumentException e)
    {
        return e.getMessage();
    }
}
```

Este método es el encargado de generar el archivo PDF con la información de la Asignación por cada Docente. Se encarga de crear el archivo, llamar a los métodos que realizan la carga de información y cerrar el archivo.

### 3.3.3.10. Clase ConReporteGrupo

Esta clase es la encargada de generar el reporte de la asignación por Docente, recibe filtros por Carrera y Nivel de la Interfaz RepGrupo.

#### Método GenerarReporteGrupo

```
public String generaReporteGrupo(String Carrera, String Nivel, String FileIN, String
FileOUT) throws DocumentException, FileNotFoundException
{
    try
    {
        Document reporte = new Document();
        PdfWriter.getInstance(reporte, new FileOutputStream(FileIN));
        reporte.setPageSize(PageSize.A4.rotate());
        reporte.setMargins(36,36,36,36);
        reporte.setMarginMirroringTopBottom(true);
        reporte.open();
        agregarContenido(reporte,Carrera, Nivel);
        reporte.close();
        gen.marcaDeAgua(FileIN,FileOUT);
        return "exito";
    }
    catch (FileNotFoundException | DocumentException e)
    {
        return e.getMessage();
    }
}
```

Este método es el encargado de generar el archivo PDF con la información de la Asignación por cada Grupo. Se encarga de crear el archivo, llamar a los métodos que realizan la carga de información y cerrar el archivo.

### 3.3.3.11. Clase ConGenerica

Esta clase es la encargada de proveer métodos de uso común a los Paquetes Vistas y Controlador, incluye a su vez los distintos tipos de fuentes que se utilizan en la generación de reportes y en las grillas de botones.

3.3.4. Paquete Modelo

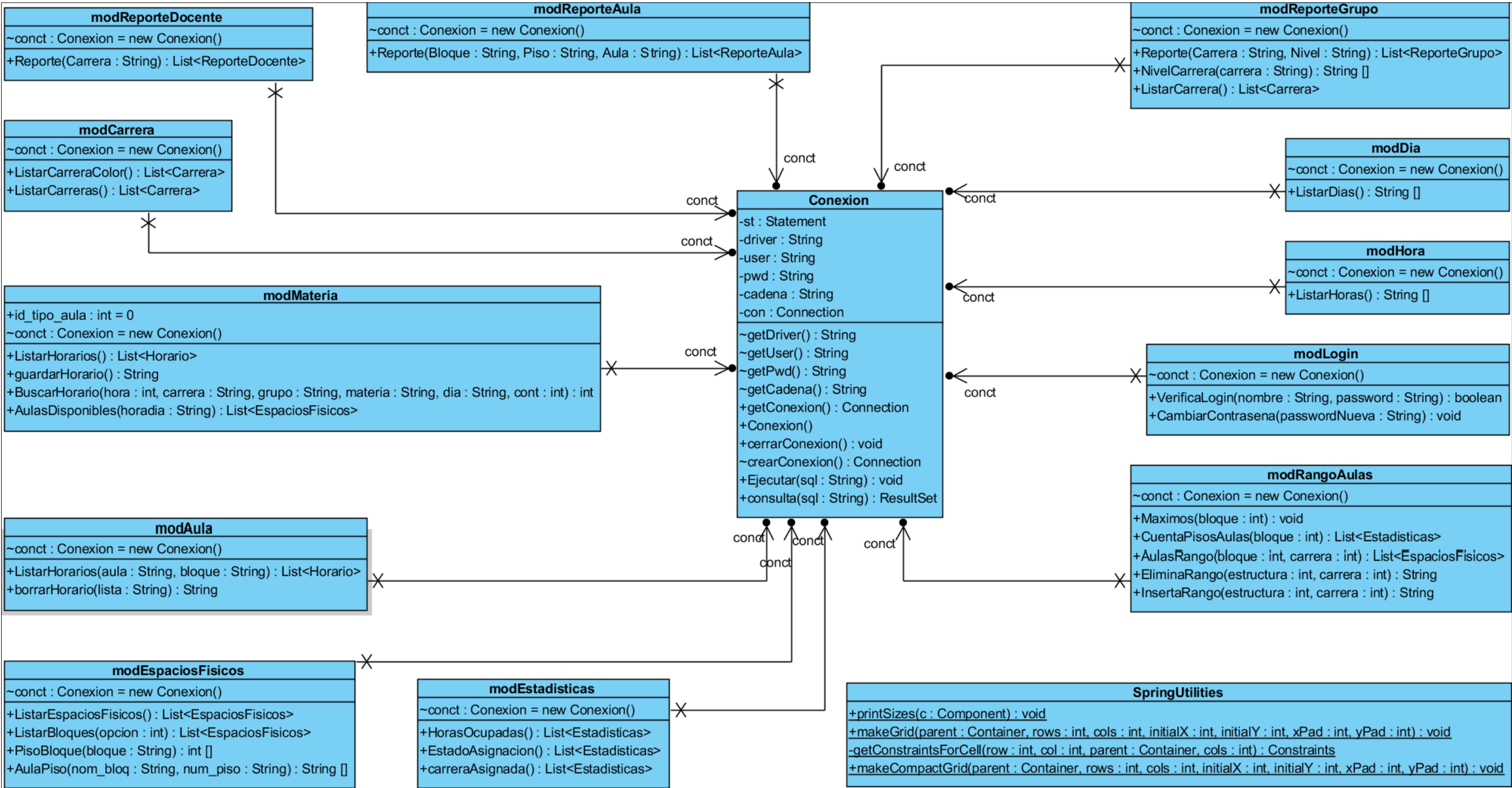


Figura 20. Diagrama de Clases del Paquete Modelo  
Elaborado por: Mauricio Romero

Este diagrama muestra las clases y las relaciones existentes dentro del paquete modelo, este paquete tiene todas las clases necesarias para realizar las consultas a la BDD, trabaja bajo petición de las clases contenidas en el paquete Controlador y utiliza los objetos definidos en el Paquete Persistencia.

#### **3.3.4.1. Clase conexión**

Contiene los métodos necesarios para:

- Establecer conexiones.
- Ejecutar consultas.
- Ejecutar afectaciones a la BDD (Insertar, Eliminar, Actualizar).
- Eliminar conexiones.

#### **3.3.4.2. Clase modLogin**

Contiene los métodos necesarios para la gestión del usuario.

- VerificaLogin: envía un script con las credenciales proporcionadas por el usuario, para realizar la validación en la BDD.
- CambiarContraseña: permite actualizar la contraseña del usuario autenticado.

#### **3.3.4.3. Clase modDia**

Contiene un Script para la extracción de los nombres de días guardados en la tabla: tbl\_dia.

#### **3.3.4.4. Clase modHora**

Contiene un Script para la extracción de los nombres de las horas guardadas en la tabla: tbl\_hora.

#### **3.3.4.5. Clase modCarrera**

Contiene un Script para la obtención de la información de todas las carreras relacionadas al campus relacionado al usuario autenticado.

#### **3.3.4.6. Clase modMateria**

Contiene un Script para la obtención de la información de todas las materias pertenecientes al campus relacionado al usuario autenticado.

#### **3.3.4.7. Clase modAula**

Contiene métodos con scripts para:

- Listar Horarios
- Eliminar asignación de aulas en horarios.

#### **3.3.4.8. Clase SpringUtilities**

Es una clase desarrollada por Oracle que no se incluye en el JDK, permite la asignación personalizada de ubicaciones de los objetos en un contenedor de tipo JPanel.

#### **3.3.4.9. Clase modEstadísticas**

Contiene métodos con Scripts para:

- Obtener el número de horarios con y sin aula asignada a nivel de campus.
- El total de horarios asignados y no asignados por carrera.
- El número de horas asignadas en cada una de las aulas de los diferentes bloques.

#### **3.3.4.10. Clase modEspaciosFisicos**

Contiene métodos con Scripts para:

- Listar los espacios físicos del campus en orden jerárquico (Campus – Sede – Bloque – Aula)
- Listar los espacios físicos de tipo bloque.
- Listar los pisos de un determinado bloque.
- Listar las aulas que se encuentran en un piso.

#### **3.3.4.11. Clase modRangoAulas**

Contiene métodos con Scripts para:

- Obtener el número máximo de aulas por piso.
- Obtener la cantidad de pisos de un bloque.
- Eliminar rango de aulas.
- Asignar rango de aulas.
- Listar rango de aulas.



#### **3.3.4.12. Clase modReporteAula**

Contiene un método con un script para la extracción de los datos requeridos para la generación del reporte de asignación por aula.

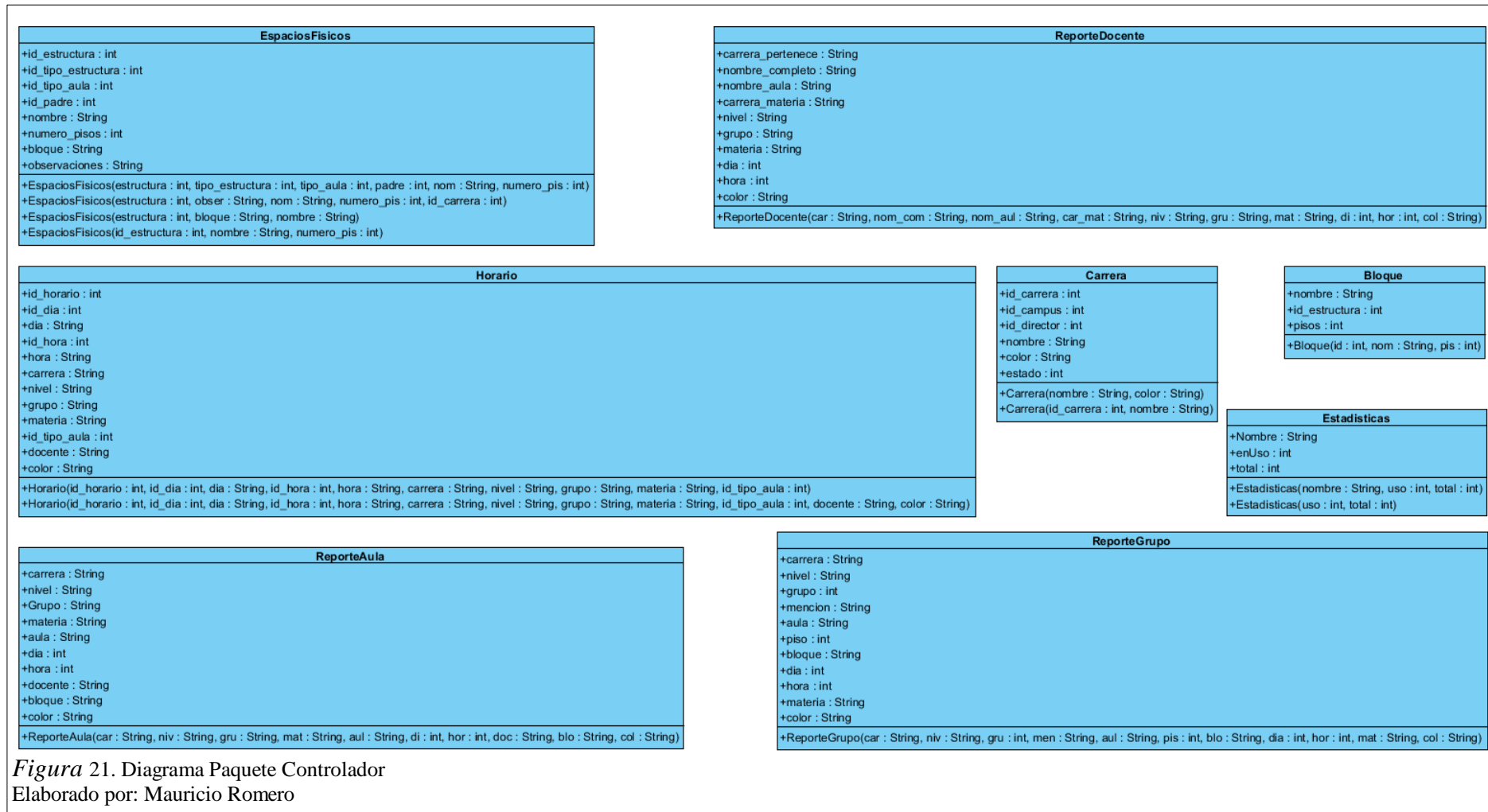
#### **3.3.4.13. Clase modReporteDocente**

Contiene un método con un script para la extracción de los datos requeridos para la generación del reporte de asignación por Docente.

#### **3.3.4.14. Clase modReporteGrupo**

Contiene un método con un script para la extracción de los datos requeridos para la generación del reporte de asignación por Docente.

### 3.3.5. Paquete Persistencia



**Figura 21.** Diagrama Paquete Controlador  
Elaborado por: Mauricio Romero

Este Paquete emula los objetos de la BDD, cada clase contiene atributos necesarios para el manejo de los resultados provenientes de las consultas realizadas a la BDD por el paquete modelo.

A continuación se muestra el uso de las clases contenidas en este paquete:

Tabla 32. *Descripción de clases del paquete Persistencia*

Clase	Objeto
Carrera	Es utilizada para almacenar el resultado del script para obtener el listado de carreras de un campus
Bloque	Es utilizada para almacenar el resultado del script para obtener el listado de los bloques de un campus
Estadísticas	Posee atributos para el manejo de los resultados obtenidos en la clase modEstadística
EspaciosFísicos	Es utilizada para guardar diferentes tipos de resultados relacionados a los espacios físicos, gracias al uso de diferentes constructores.
Horario	Se utiliza para almacenar los resultados de los scripts relacionados a la obtención de los horarios del campus.
ReporteAula	Contiene los atributos necesarios para el manejo del resultado del script para la generación del Reporte de Asignación por Aula.
ReporteDocente	Contiene los atributos necesarios para el manejo del resultado del script para la generación del Reporte de Asignación por Docente.
ReporteGrupo	Contiene los atributos necesarios para el manejo del resultado del script para la generación del Reporte de Asignación por Grupo.

Nota.

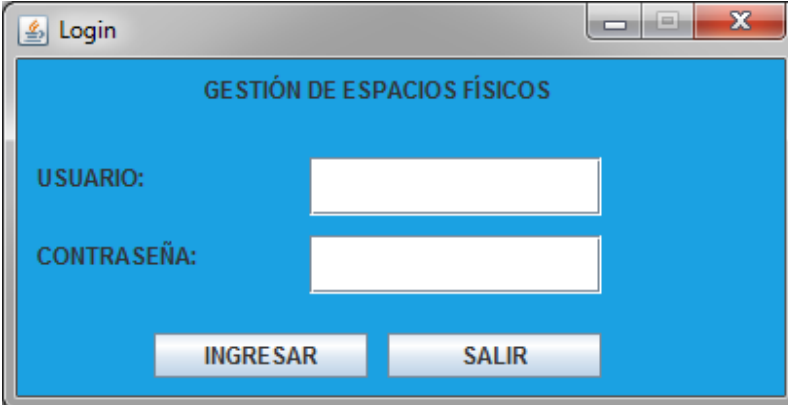
Elaborado por: Mauricio Romero

### 3.4. Resultados

A continuación se presentan las interfaces más importantes del Sistema, para observar el detalle del funcionamiento de toda la aplicación.

#### 3.4.1. Interfaz Login

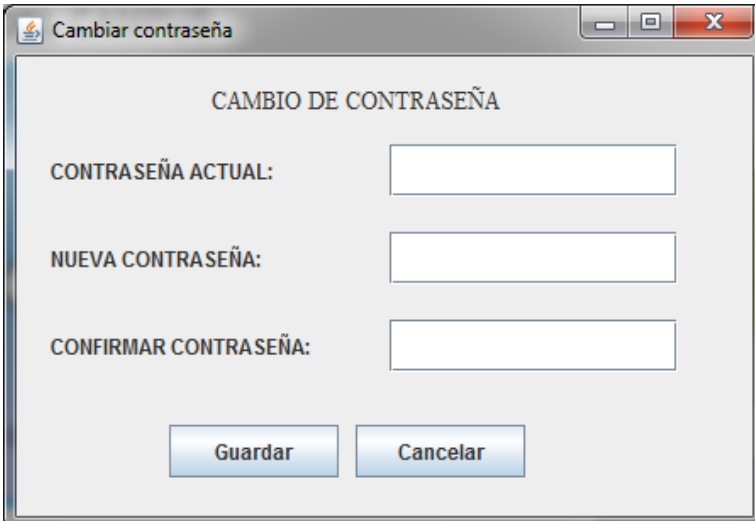
Resultado



*Figura 22.* Interfaz Login – Sistema Gestión de Espacios Físicos  
Elaborado por: Mauricio Romero

#### 3.4.2. Interfaz Cambiar Contraseña

Resultado



*Figura 23.* Interfaz Cambiar Contraseña – Sistema Gestión de Espacios Físicos  
Elaborado por: Mauricio Romero

### 3.4.3. Interfaz Inicio

Resultado

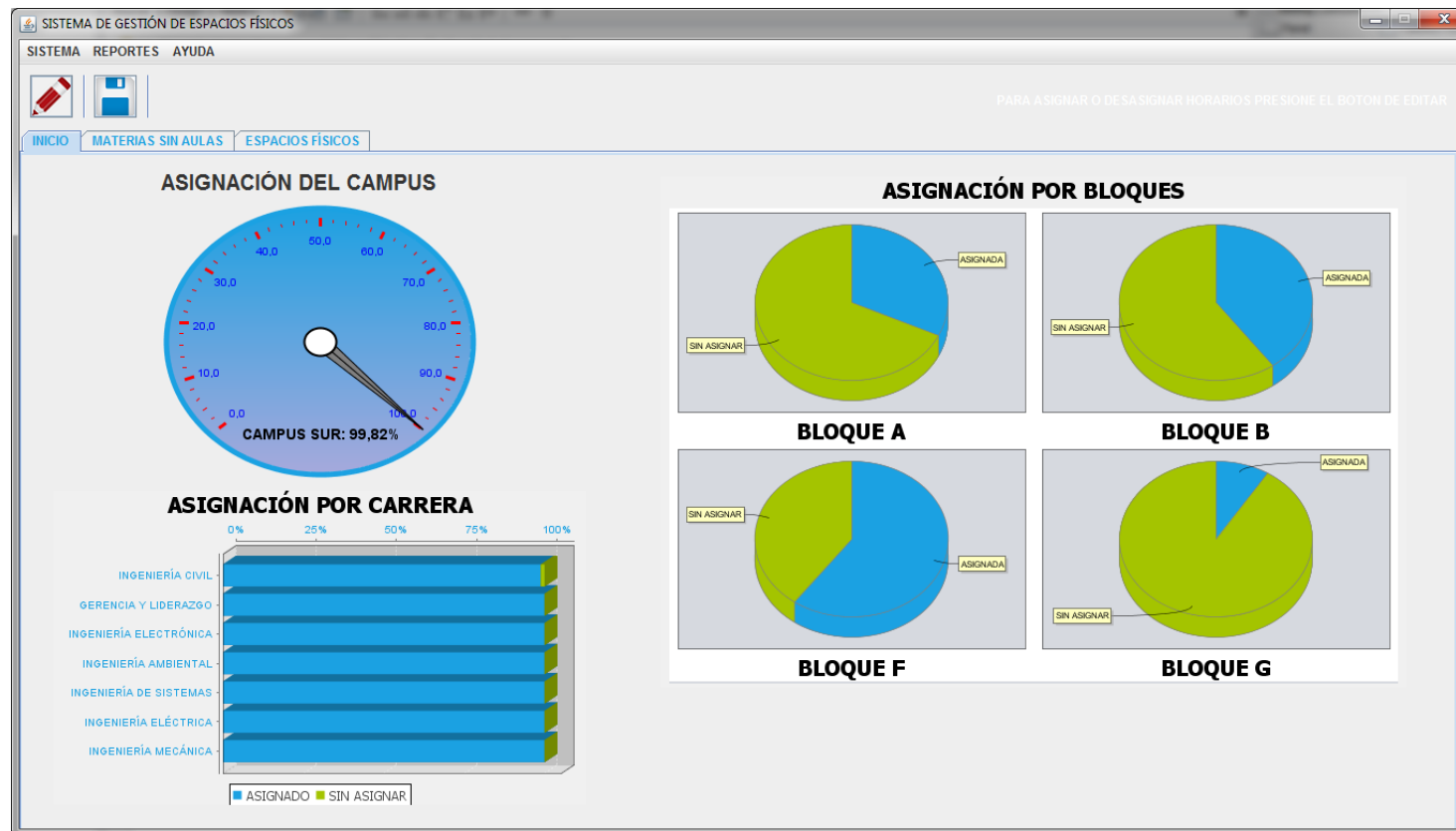


Figura 24. Interfaz Inicio – Sistema Gestión de Espacios Físicos

Elaborado por: Mauricio Romero

### 3.4.4. Interfaz Materias sin aulas

Resultado

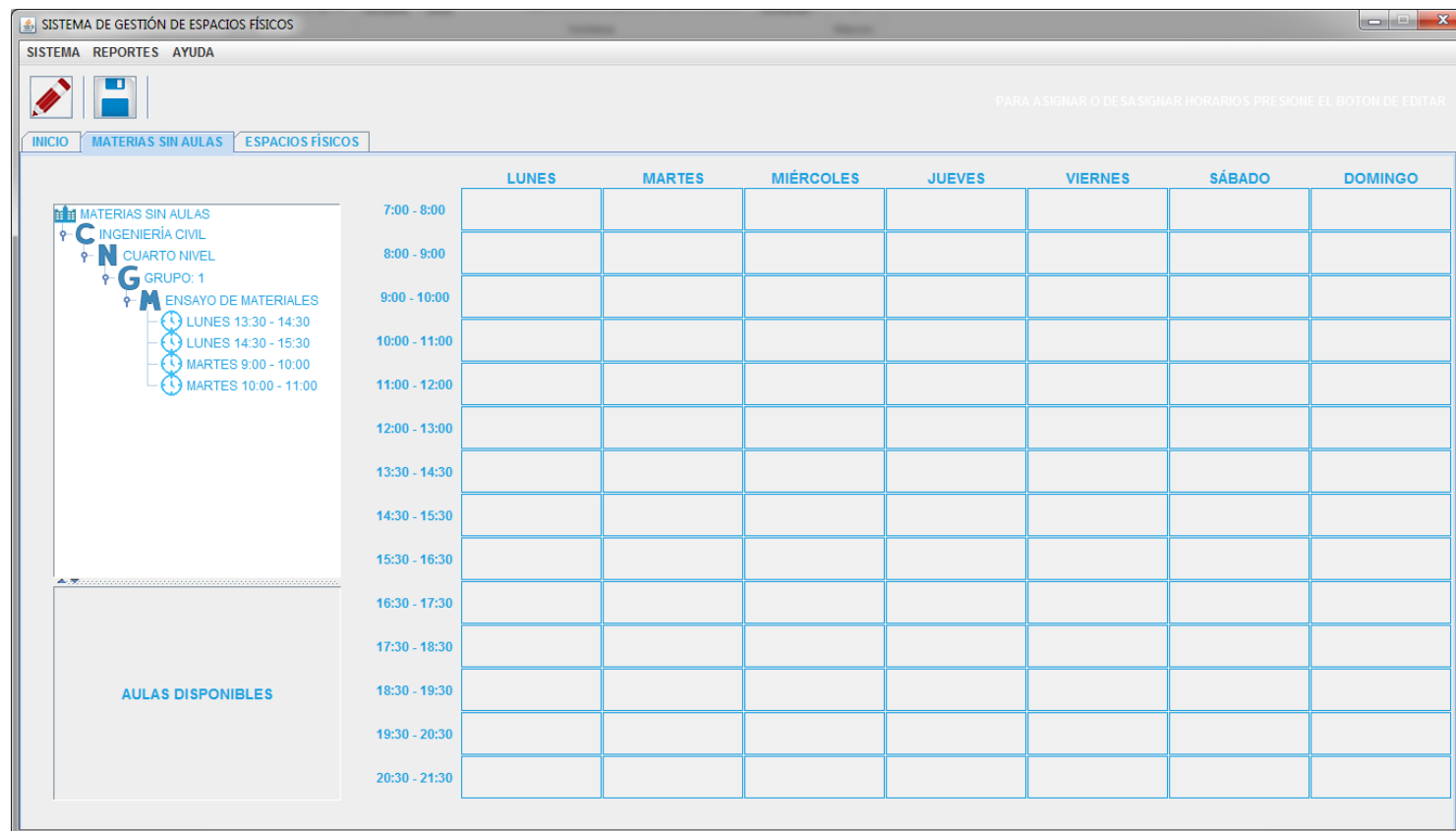


Figura 25. Interfaz Materias sin aulas – Sistema Gestión de Espacios Físicos

Elaborado por: Mauricio Romero

3.4.5. Interfaz Espacios Físicos

Resultado




Figura 26. Interfaz Espacios Físicos – Sistema Gestión de Espacios Físicos

Elaborado por: Mauricio Romero

### 3.4.6. Reporte por Aula

Resultado

BLOQUE A



P46

A1

HORA	LUNES		MARTES		MIÉRCOLES		JUEVES		VIERNES		SABADO							
07:00	IS	<div style="border: 1px solid black; padding: 2px;">II G: 2</div>	CALCULO INTEGRAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	TÉCNICAS DE INVESTIGACIÓN	IS	<div style="border: 1px solid black; padding: 2px;">II G: 3</div>	ANTROPOLOGÍA CRISTIANA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	DEONTOLOGÍA	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INGENIERÍA DEL TRÁFICO (T)
08:00	IS	<div style="border: 1px solid black; padding: 2px;">II G: 2</div>	CALCULO INTEGRAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	TÉCNICAS DE INVESTIGACIÓN	IS	<div style="border: 1px solid black; padding: 2px;">II G: 3</div>	ANTROPOLOGÍA CRISTIANA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	DEONTOLOGÍA	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INGENIERÍA DEL TRÁFICO (T)
09:00	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	ÁLGEBRA LINEAL	IS	<div style="border: 1px solid black; padding: 2px;">III G: 1</div>	ESPIRITUALIDAD JUVENIL SALESIANA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">VI G: 2</div>	ÉTICA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	DISEÑO DE REDES (T)
10:00	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	ÁLGEBRA LINEAL	IS	<div style="border: 1px solid black; padding: 2px;">III G: 1</div>	ESPIRITUALIDAD JUVENIL SALESIANA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">VI G: 2</div>	ÉTICA	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	DISEÑO DE REDES (T)
11:00	IS	<div style="border: 1px solid black; padding: 2px;">II G: 1</div>	CALCULO INTEGRAL	IS	<div style="border: 1px solid black; padding: 2px;">IV G: 1</div>	MATEMÁTICAS SUPERIORES	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	ÁLGEBRA LINEAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL			
12:00	IS	<div style="border: 1px solid black; padding: 2px;">II G: 1</div>	CALCULO INTEGRAL	IS	<div style="border: 1px solid black; padding: 2px;">IV G: 1</div>	MATEMÁTICAS SUPERIORES	IS	<div style="border: 1px solid black; padding: 2px;">I G: 3</div>	CALCULO DIFERENCIAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	ÁLGEBRA LINEAL	IS	<div style="border: 1px solid black; padding: 2px;">I G: 1</div>	CALCULO DIFERENCIAL			
13:30																		
14:30																		
15:30																		
16:30																		
17:30	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	CONTROL DE CALIDAD (O)	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INVESTIGACIÓN DE OPERACIONES	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	ADMINISTRACIÓN Y GESTIÓN DE REDES (T)	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	DISEÑO DE REDES (T)	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	SISTEMAS OPERACIONALES (O)			
18:30	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	CONTROL DE CALIDAD (O)	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INVESTIGACIÓN DE OPERACIONES	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	ADMINISTRACIÓN Y GESTIÓN DE REDES (T)	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	DISEÑO DE REDES (T)	IS	<div style="border: 1px solid black; padding: 2px;">X G: 1</div>	SISTEMAS OPERACIONALES (O)			
19:30					IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	SISTEMAS EXPERTOS (O)	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INVESTIGACIÓN DE OPERACIONES	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	SISTEMAS EXPERTOS (O)	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	ADMINISTRACIÓN Y GESTIÓN DE REDES (T)		
20:30					IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	SISTEMAS EXPERTOS (O)	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	INVESTIGACIÓN DE OPERACIONES	IS	<div style="border: 1px solid black; padding: 2px;">XI G: 1</div>	SISTEMAS EXPERTOS (O)	IS	<div style="border: 1px solid black; padding: 2px;">VIII G: 1</div>	ADMINISTRACIÓN Y GESTIÓN DE REDES (T)		

IS

 INGENIERÍA DE SISTEMAS

Figura 27. Reporte de aula – Sistema Gestión de Espacios Físicos


Elaborado por: Mauricio Romero



### 3.4.7. Reporte por Docente

Resultado

VILLAGOMEZ MENENDEZ JOSE LUIS



P46

HORA	LUNES			MARTES			MIÉRCOLES			JUEVES			VIERNES	SABADO
07:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
08:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
09:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
10:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
11:00				L2	II 1	PROGRAMACION II								
12:00				L2	II 1	PROGRAMACION II								
13:30														
14:30														
15:30														
16:30														
17:30														
18:30														
19:30														
20:30														

IS


INGENIERÍA DE SISTEMAS

*Figura 28.* Reporte de docente – Sistema Gestión de Espacios Físicos  
Elaborado por: Mauricio Romero

3.4.8. Reporte por Grupo

Resultado

VILLAGOMEZ MENENDEZ JOSE LUIS

UNIVERSIDAD POLITÉCNICA  
SALESIANA  
ECUADOR

P46

HORA	LUNES			MARTES			MIÉRCOLES			JUEVES			VIERNES	SABADO
07:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
08:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
09:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
10:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
11:00				L2	II 1	PROGRAMACION II								
12:00				L2	II 1	PROGRAMACION II								
13:30														
14:30														
15:30														
16:30														
17:30														
18:30														
19:30														
20:30														

IS INGENIERÍA DE SISTEMAS

Figura 29. Reporte de grupo – Sistema Gestión de Espacios Físicos  
Elaborado por: Mauricio Romero

## CAPÍTULO 4

### 4. CARGA DE DATOS Y PRUEBAS

En este capítulo se detalla la carga inicial de las tablas necesarias para el funcionamiento del sistema.

Se presenta el contenido que deben tener las tablas de tipo catalogo y se describe el proceso de carga de los archivos procedentes del Sistema SNA a la Base de Datos UPSSchedule, además se muestran los resultados de la encuesta realizada a los usuarios que probó el Programa.

#### 4.1. Carga de información a la base de datos UPSSchedule

Para realizar las pruebas de aceptación con el Usuario final, se procedió con la carga de la información del Periodo 46.

##### 4.1.1. Tablas catálogo

Las siguientes tablas, contienen información definida y aceptada por todos los módulos del Proyecto UPSSchedule, deben ser cargadas con la información que se muestra a continuación.

Tabla 33. *Contenido de la tabla: tbl\_tipo\_estructura*

tbl_tipo_estructura	
Id_Tipo_Estructura	Nombre
1	UNIVERSIDAD
2	SEDE
3	CAMPUS
4	BLOQUE
5	AULA

Nota.

Elaborado por: Mauricio Romero

Tabla 34. *Contenido de la tabla: tbl\_tipo\_aula*

tbl_tipo_aula		
Id_Tipo_Materia	Nombre	Observaciones
0	NO DEFINIDO	1
1	REGULAR	1
2	LABORATORIO CECASI	1

Nota.

Elaborado por: Mauricio Romero

Tabla 35. *Contenido de la tabla: tbl\_estadoh*

tbl_estadoh			
Id_Estadoh	Nombre	Estado	Color
1	Disponible	1	#DFFBE4
2	Cruce horario	1	#EDECAC
3	Cruce otra	1	#5C5C71
4	No disponible	1	#B15252

Nota.

Elaborado por: Mauricio Romero

Tabla 36. *Contenido de la tabla: tbl\_dia*

tbl_dia		
Id_Dia	Dia	Inicial
0	SIN ASIGNAR	NE
1	LUNES	L
2	MARTES	MA
3	MIERCOLES	MI
4	JUEVES	J
5	VIERNES	V
6	SABADO	S
7	DOMINGO	D

Nota.

Elaborado por: Mauricio Romero

Tabla 37. *Contenido de la tabla: tbl\_hora*

tbl_hora		
Id_Hora	Inicio	Fin
0	0	0
1	7	8
2	8	9
3	9	10
4	10	11
5	11	12
6	12	13
7	13	14
8	14	15
9	15	16
10	16	17
11	17	18
12	18	19
13	19	20
14	20	21

Nota.

Elaborado por: Mauricio Romero

Tabla 38. *Contenido de la tabla: tbl\_periodo*

tbl_periodo			
Id_Periodo	Numero	Descripcion	Estado
1	46	ABRIL 2015 – JULIO 2015	1

Nota.

Elaborado por: Mauricio Romero

Tabla 39. *Contenido de la tabla: tbl\_malla*

tbl_malla			
Id_Malla	Id_Carrera	Nombre	Estado
1	1	VIGENTE	1
2	2	VIGENTE	1
3	3	VIGENTE	1
4	4	VIGENTE	1
5	5	VIGENTE	1
6	6	VIGENTE	1
7	7	VIGENTE	1

Nota.

Elaborado por: Mauricio Romero

Tabla 40. *Contenido de la tabla: tbl\_nivel*

tbl_nivel		
Id_Nivel	Numero	Nombre
0	0	PROPEDEUTICO
1	1	PRIMERO
2	2	SEGUNDO
3	3	TERCERO
4	4	CUARTO
5	5	QUINTO
6	6	SEXTO
7	7	SEPTIMO
8	8	OCTAVO
9	9	NOVENO
10	10	DECIMO

Nota.

Elaborado por: Mauricio Romero

Tabla 41. *Contenido de la tabla: tbl\_tipo\_estructura*

tbl_tipo_estructura	
Id_Tipo_Estructura	Nombre
1	SIN MENCION
2	GESTIÓN
3	TELEMÁTICA
4	SISTEMAS INDUSTRIALES
5	TELECOMUNICACIONES

Nota.

Elaborado por: Mauricio Romero

#### 4.1.2. Archivos UPS

La Universidad Politécnica Salesiana proveyó de los siguientes archivos, con información proveniente del SNA.

Tabla 42. *Descripción archivo Distributivo*

Archivo: Distributivo	
SEDE	Nombre de la Sede a la cual pertenece la materia.
CAMPUS	Nombre del Campus al cual pertenece la materia.
FACULTAD	Nombre de la Facultad a la cual pertenece la materia.
CARRERA	Nombre de la Carrera a la cual pertenece la materia.
CEDULA	Número de cédula del Docente responsable de la materia.
DOCENTE	Nombre del Docente responsable de la materia.
CODIGO	Código interno (SNA) de la materia.
MATERIA	Nombre de la materia.
NOMBRE_GRUPO	Nombre del grupo.
HORAS	Número de horas.
TIPO	Tipo de Materia.
PERIODO	Descripción del Periodo.
TIPO_PARACADEMICO	Tipo de paracadémico al cual pertenece la materia.

Nota.

Elaborado por: Mauricio Romero

Tabla 43. *Descripción archivo Materias Ofertadas*

Archivo: Materias Ofertadas	
PERIODO	Descripción del Periodo.
SEDE	Nombre de la Sede a la cual pertenece la materia.
CAMPUS	Nombre del Campus al cual pertenece la materia.
FACULTAD	Nombre de la Facultad a la cual pertenece la materia.
CARRERA	Nombre de la Carrera a la cual pertenece la materia.
MAT_CODIGO	Código interno (SNA) de la materia.
MAT_DESCRIPCION	Nombre de la materia.
NIVEL	Número de Nivel al cual pertenece la materia.
TIPO	Tipo de Materia.

Nota.

Elaborado por: Mauricio Romero

Tabla 44. *Descripción archivo Horario*

Archivo: Horario	
PERIODO	Descripción del Periodo.
SEDE	Nombre de la Sede a la cual pertenece la materia.
CAMPUS	Nombre del Campus al cual pertenece la materia.
FAC_DESCRIPCION	Nombre de la Facultad a la cual pertenece la materia.
CARRERA	Nombre de la Carrera a la cual pertenece la materia.
CEDULA	Número de cédula del Docente responsable de la materia.
DOCENTE	Nombre del Docente responsable de la materia.
CODIGO_DISTRIBUTIVO	Campo que identifica el distributivo al cual pertenece el horario (columna vacía).

CODIGO	Código interno (SNA) de la materia.
MATERIA	Nombre de la materia.
NIVEL	Número de Nivel al cual pertenece la materia.
GRUPO	Nombre del grupo.
HORA_INICIAL	Hora en la que inicia la clase.
HORA_FINAL	Hora en la que termina la clase.
DIA	Día en el que se dicta la clase.
FECHA_INICIAL	Fecha desde la cual se dicta la clase.
FECHA_FINAL	Fecha hasta cuando se dicta la clase.
TIPO	Tipo de Materia.
HORAS	Números de horas
PERIODO	Descripción del Periodo.
TIPO_PARACADEMICO	Tipo de paracadémico al cual pertenece la materia.

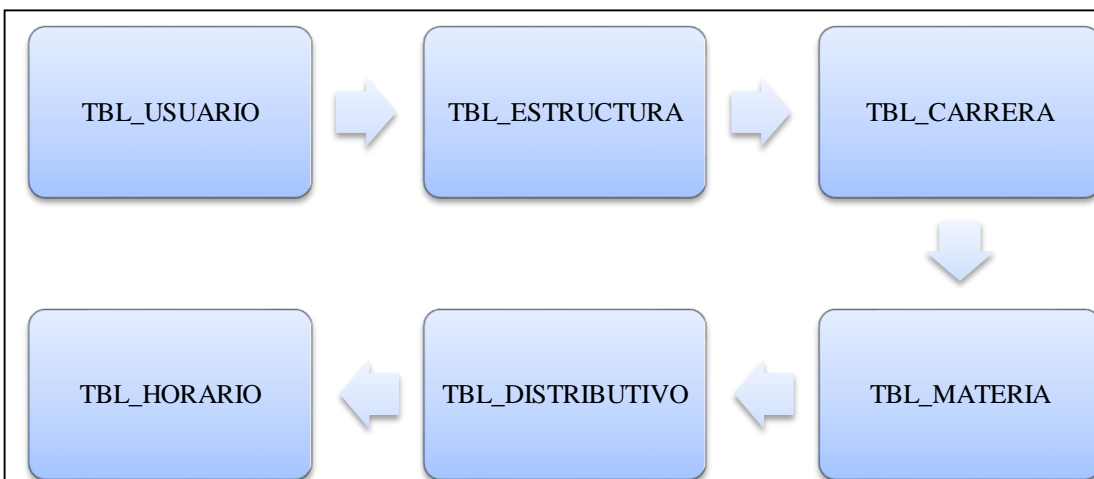
Nota.

Elaborado por: Mauricio Romero

#### 4.1.3. Proceso de carga

A continuación se detalla el proceso de carga de la información proveniente de los Archivos UPS, durante el proceso de carga se hace referencia a tablas de la BDD UPSSCHEDULE, esto significa que la información de esas tablas debe ser incluida en los archivos de Excel en los cuales se realiza el proceso.

Una vez cargadas las tablas catálogo, se procede a cargar las siguientes tablas según el esquema presentado a continuación



*Figura 30.* Esquema carga tablas BDD UPSSchedule

Elaborado por: Mauricio Romero

### 4.1.3.1. Carga tbl\_usuario

#### Proceso de Carga

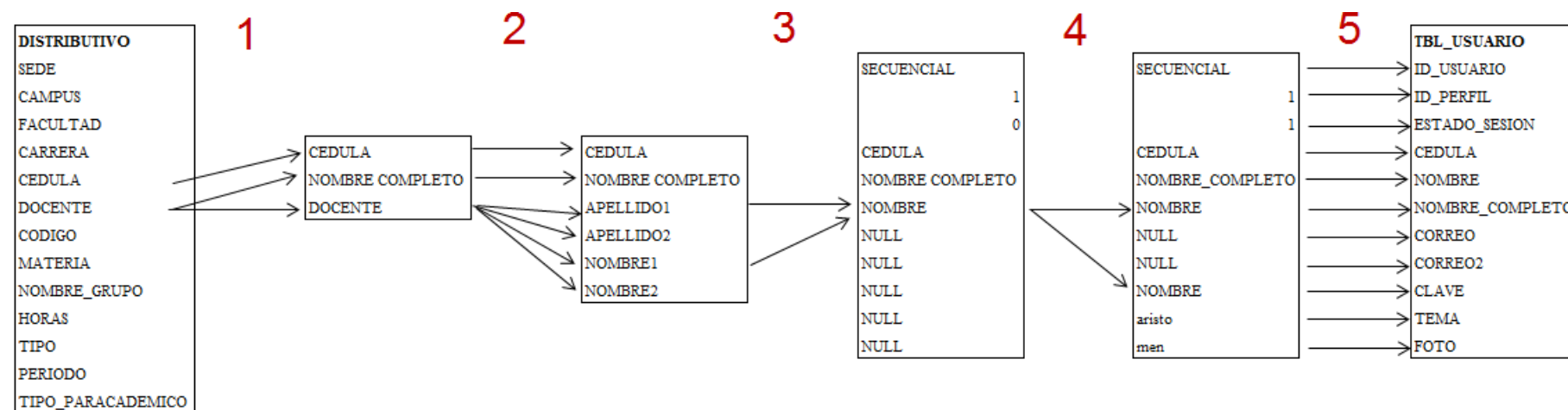


Figura 31. Proceso de carga de tbl\_usuario

Elaborado por: Mauricio Romero

- 1) Archivo Distributivo proporcionado por la UPS.  
Extraer los datos de las columnas cedula y docente.  
Borrar registros duplicados con la función "Quitar duplicados"
- 2) Aplicar función "Texto en columnas" al campo docente.  
Corregir los casos en los que se generan más de 4 columnas por usuario.
- 3) Aplicar la función MED para extraer la primera letra de la columna nombre1 y unirla con la columna apellido1 mediante la función CONCATENAR.

Limitar a 10 caracteres la nueva columna mediante la función MED, y reemplazar los caracteres especiales a través de la función SUSTITUIR.  
Validar que no existan valores duplicados en la columna nombre

- 4) Duplicar la columna nombre.  
Asignar valores enteros secuenciales a los registros.  
Crear columnas con valores nulos o valores por defecto.

Para visualizar el formato destino, ver Anexo 1.



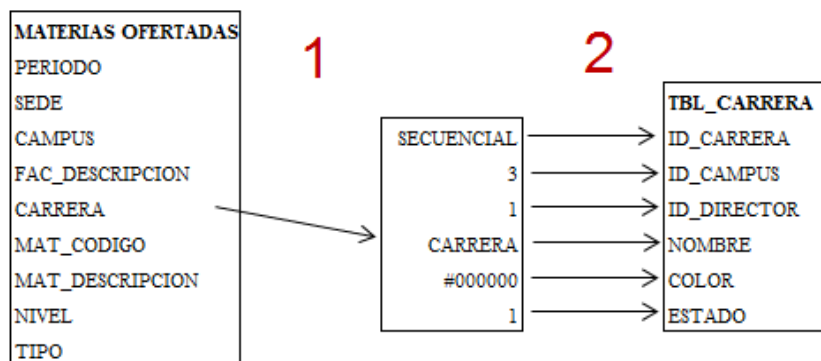
#### 4.1.3.2. Carga tbl\_estructura

El proceso de carga de tabla Tbl\_estructura, fue realizado de forma manual en base al conocimiento previo sobre la infraestructura y la información brindada por la Ingeniera Patricia Quiroz Administradora de los Espacios Físicos en el Campus Sur.

Para visualizar el formato destino, ver Anexo 2.

#### 4.1.3.3. Carga tbl\_carrera

Proceso de carga



*Figura 32.* Proceso de carga de tbl\_carrera  
Elaborado por: Mauricio Romero

- 1) Archivo **Materias Ofertadas** proporcionado por la UPS.  
Extraer los datos de la columna carrera.  
Borrar registros duplicados con la función “**Quitar duplicados**”
- 2) Asignar valores enteros secuenciales a los registros.

Crear columnas con valores por defecto.

Registrar el código asignado a cada director de carrera.

Registrar el código asignado al Campus al cual pertenece la materia

Para visualizar el formato destino, ver Anexo 3.

#### 4.1.3.4. Carga tbl\_materia

##### Proceso de Carga

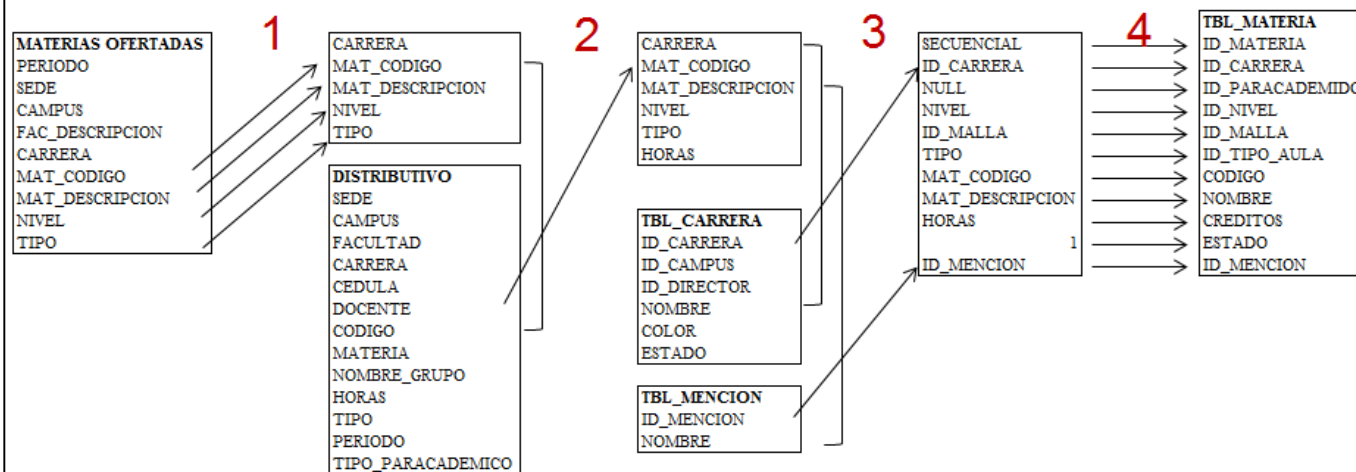


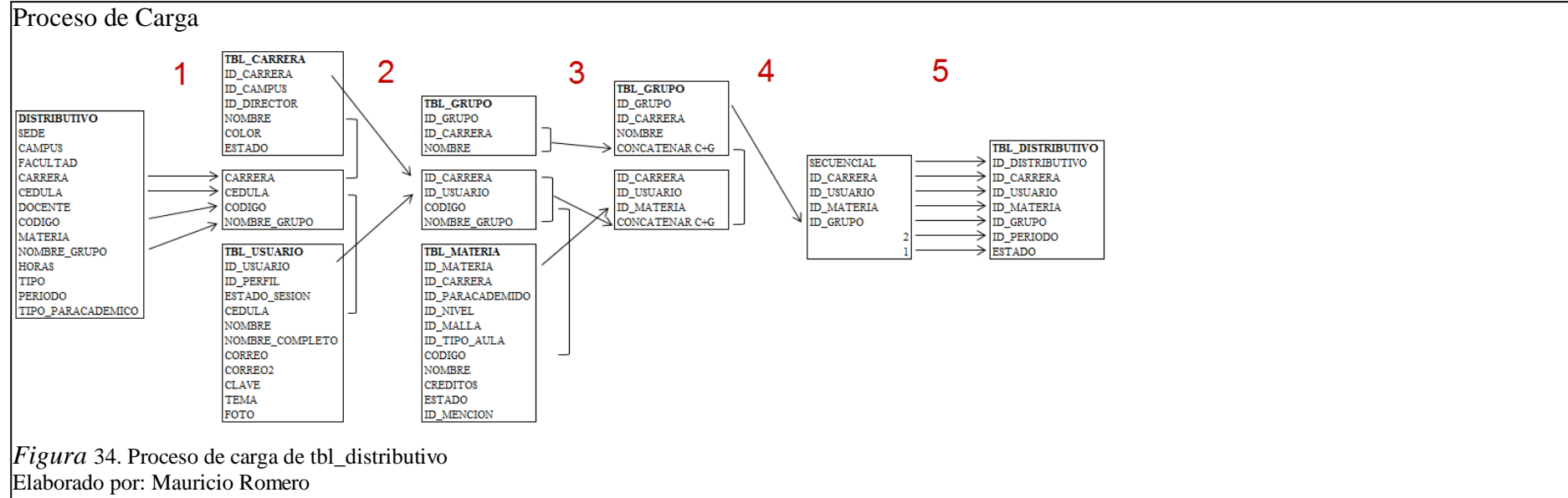
Figura 33. Proceso de carga de tbl\_materia

Elaborado por: Mauricio Romero

- 1) Archivo **Materias Ofertadas** proporcionado por la UPS.  
Extraer los datos de las columnas MAT\_CODIGO, MAT\_DESCRIPCION, NIVEL, TIPO  
Borrar registros duplicados con la función “**Quitar duplicados**”  
  
Analizar los nombres de materias que hacen referencia a una mención (P. ej. (T) → Telemática) y asignarles el ID\_MENCION correspondiente de la tabla TBL\_MENCION.
- 2) Obtener la columna Horas del Archivo **Distributivo**, relacionando las columnas MAT\_CODIGO y CODIGO mediante la función **CONSULTARV**.
- 3) Obtener la columna ID\_CARRERA, relacionando las columnas CARRERA y NOMBRE mediante la función **CONSULTARV**.  
Asignar valores enteros secuenciales a los registros.  
Crear columnas con valores por defecto.

Para visualizar el formato destino, ver Anexo 4.

#### 4.1.3.5. Carga tbl\_Distributivo



- 1) Archivo **Distributivo** proporcionado por la UPS.  
Extraer los datos de las columnas: carrera, cedula, codigo, nombre\_grupo.
- 2) Obtener la columna **id\_carrera** de la tabla **tbl\_carrera** relacionando las columnas **Carrera** y **Nombre** mediante la función **CONSULTARV**.  
Obtener la columna **id\_usuario** de la tabla **tbl\_usuario** relacionando las columnas **cedula** y **cedula** mediante la función **CONSULTARV**.
- 3) **CONCATENAR** con las columnas **id\_carrera** y **nombre**, y las columnas **id\_carrera** y **grupo**

Obtener la columna **id\_materia** de la tabla **tbl\_materia** relacionando las columnas **codigo** y **codigo** mediante la función **CONSULTARV**.

- 4) Obtener la columna **id\_grupo** de la tabla **tbl\_grupo** relacionando las columnas **concatenar c+g** y **concatenar c+g** mediante la función **CONSULTARV**.

Asignar valores enteros secuenciales a los registros.

Crear columnas con valores por defecto o nulos.

Para visualizar el formato destino, ver Anexo 5.

#### 4.1.3.6. Carga tbl\_horario

Proceso de carga

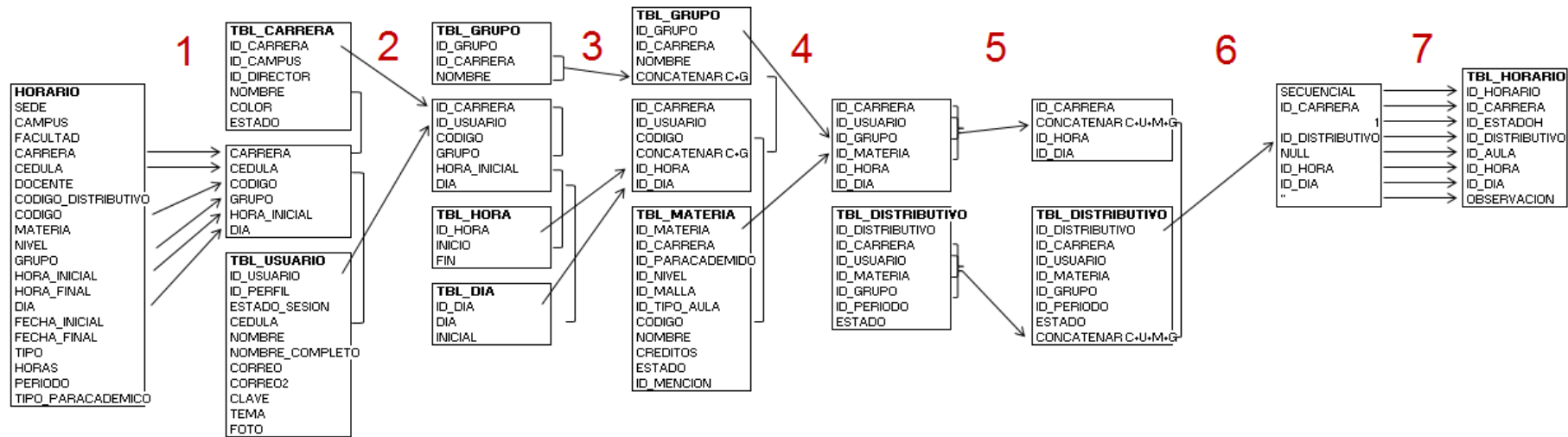


Figura 35. Proceso de carga de la tbl\_horario  
Elaborado por: Mauricio Romero

- 1) Archivo **Materias Ofertadas** proporcionado por la UPS.  
Extraer los datos de las columnas: carrera, cedula, codigo, grupo, hora Inicial y dia.
- 2) Obtener la columna id\_carrera de la tabla tbl\_carrera relacionando las columnas Carrera y Nombre mediante la función **CONSULTARV**.  
Obtener la columna id\_usuario de la tabla tbl\_usuario relacionando las columnas cedula y cedula mediante la función **CONSULTARV**.
- 3) Crear columnas “CONCATENAR C+G” mediante el uso de la función **CONCATENAR** con las columnas id\_carrera y nombre, y las columnas id\_carrera y grupo.  
Obtener la columna id\_hora de la tabla tbl\_hora relacionando las columnas inicio y hora\_inicial mediante la función **CONSULTARV**.
- 4) Obtener la columna id\_grupo de la tabla tbl\_grupo relacionando las columnas concatenar c+g y concatenar c+g mediante la función **CONSULTARV**.  
Obtener la columna id\_materia de la tabla tbl\_materia relacionando las columnas codigo y codigo mediante la función **CONSULTARV**.
- 5) Crear columnas “concatenar c+u+m+g” mediante el uso de la función **CONCATENAR** con las columnas id\_carrera, id\_usuario, id\_materia e id\_grupo.
- 6) Obtener la columna id\_distributivo de la tabla tbl\_distributivo relacionando las columnas concatenar c+u+m+g y concatenar c+u+m+g mediante la función **CONSULTARV**.  
Asignar valores enteros secuenciales a los registros.  
Crear columnas con valores por defecto o nulos.

Para visualizar el formato destino, ver Anexo 6.

## 4.2. Pruebas

A continuación se muestran los resultados de las prueba contra requerimientos y pruebas de usabilidad a las que se sometió el programa.

### 4.2.1. Pruebas contra requerimientos

A continuación se presentan los resultados de la prueba contra requerimientos realizada al usuario final del Programa.

Tabla 45. *Pruebas contra requerimientos*

N°	Nombre de la Prueba	Req.	Procedimiento	Estado	Observaciones
1	Acceder al Sistema	1.1	1. Abrir el programa 2. Ingresar Usuario 3. Ingresar Contraseña 4. Presionar el botón Ingresar	OK	
2	Modificar clave de Acceso	1.2	1. Acceder al sub menú “Cambio de Contraseña” o Presionar Ctrl + U. 2. Ingresar contraseña Actual. 3. Ingresar nueva contraseña. 4. Ingresar confirmación de la nueva contraseña 5. Presionar el botón Guardar	OK	
3	Visualizar Estadísticas de Asignación	1.3	1. Acceder a la pestaña “Inicio” o Presionar Ctrl + I. 2. Observar los siguientes cuadros estadísticos: a) Estado Global de la Asignación. b) Estado de la Asignación por Carrera. c) Estado de la Asignación por Bloques.	OK	
4	Gestión materias sin aula	1.4	1. Acceder a la pestaña “Gestión de Aulas” o Presionar Ctrl + M. 2. Seleccionar una Carrera. 3. Seleccionar un Nivel. 4. Seleccionar un Grupo. 5. Seleccionar horario o grupo de horarios por materia. 6. Visualizar aulas disponibles para la solicitud realizada. 7. Seleccionar aula disponible. 8. Presionar el Botón de Guardar.	OK	
5	Eliminar Asignación de Aula	1.5	1. Acceder a la pestaña “Espacios Físicos” o Presionar Ctrl + F. 2. Seleccionar Bloque. 3. Seleccionar Piso. 4. Seleccionar Aula. 5. Presionar Botón de Edición. 6. Seleccionar horario u horarios a eliminar. 7. Presionar botón de Guardar.	OK	

			8. Confirmar la eliminación.		
6	Generación de Reporte por Docente	1.6	1. Acceder al sub menú de Reportes o Presionar Ctrl + R. 2. En el panel “REPORTE DE DOCENTES” seleccionar una Carrera. 3. Elegir nombre y ruta donde se guardará el archivo. 4. Visualizar Reporte	NO	El tamaño de la fuente para el nombre del Aula es demasiado grande.
7	Generación de Reporte por Grupo	1.7	1. Acceder al sub menú de Reportes o Presionar Ctrl + R. 2. En el panel “REPORTE DE GRUPOS” seleccionar una Carrera. 3. Seleccionar un Nivel. 4. Elegir nombre y ruta donde se guardará el archivo. 5. Visualizar Reporte	OK	
8	Generación de Reporte por Aula	1.8	1. Acceder al sub menú de Reportes o Presionar Ctrl + R. 2. En el panel “REPORTE DE AULAS” seleccionar una Carrera. 3. Seleccionar un Piso. 4. Seleccionar un Aula. 5. Elegir nombre y ruta donde se guardará el archivo. 6. Visualizar Reporte	OK	

Nota.

Elaborado por: Mauricio Romero

#### 4.2.2. Pruebas de usabilidad

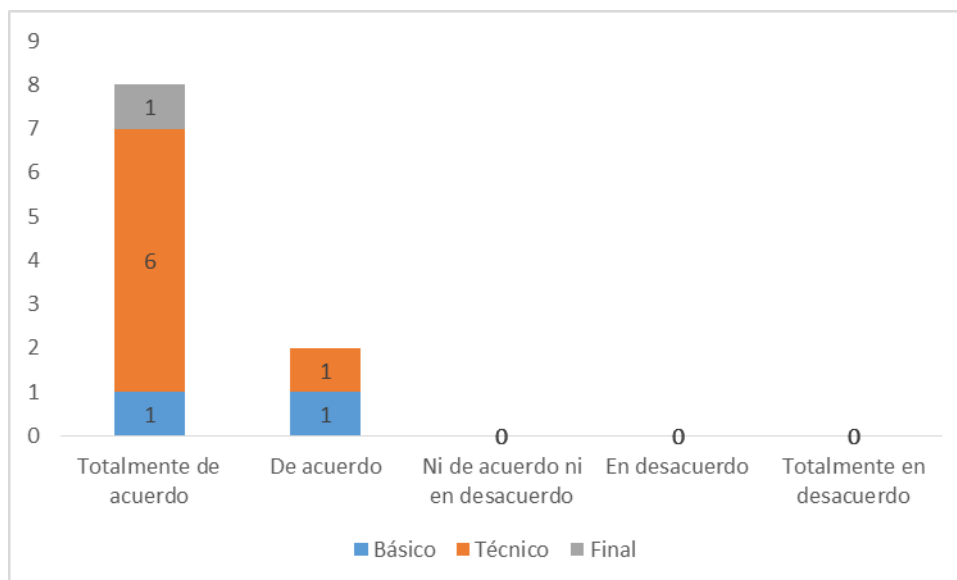
Estas pruebas se realizan con la finalidad de evaluar la facilidad de uso de los sistemas, los datos obtenidos permiten realizar mejoras, correcciones a las dificultades, errores encontrados por los diferentes usuarios.

Estas pruebas fueron realizadas por 10 usuarios que manipularon el sistema en su totalidad, a continuación se detallan los perfiles de los usuarios.

- 1 Usuario final – Asistente Administrativo.
- 7 Usuarios con perfil técnico (Ingenieros de Sistemas).
- 2 Usuarios con conocimientos básicos de informática.

1. ¿Considera usted, que la información estadística mostrada, es clara y tiene el suficiente nivel de detalle?

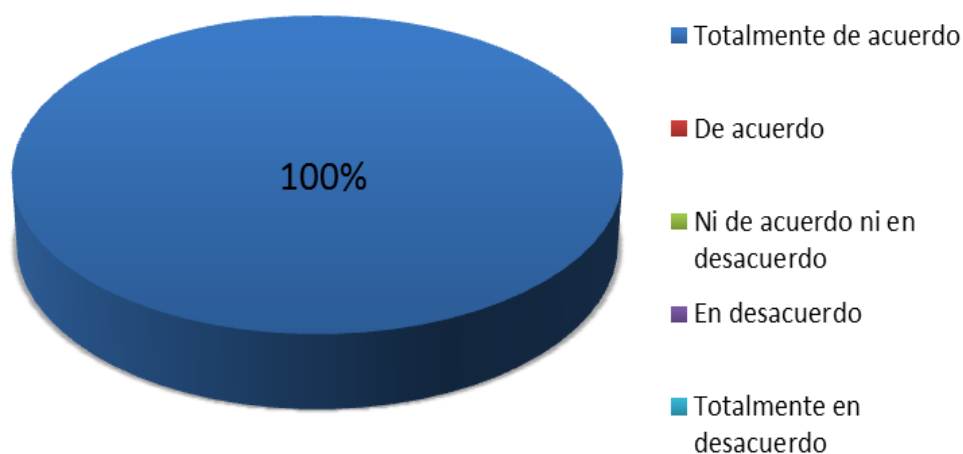
#### Resultados



*Figura 36.* Pregunta 1 – Prueba de Usabilidad  
Elaborado por: Mauricio Romero

2. ¿Considera usted, qué la interfaz de Materias sin Aulas permite una fácil búsqueda y asignación de materias que no tienen un aula asignada?

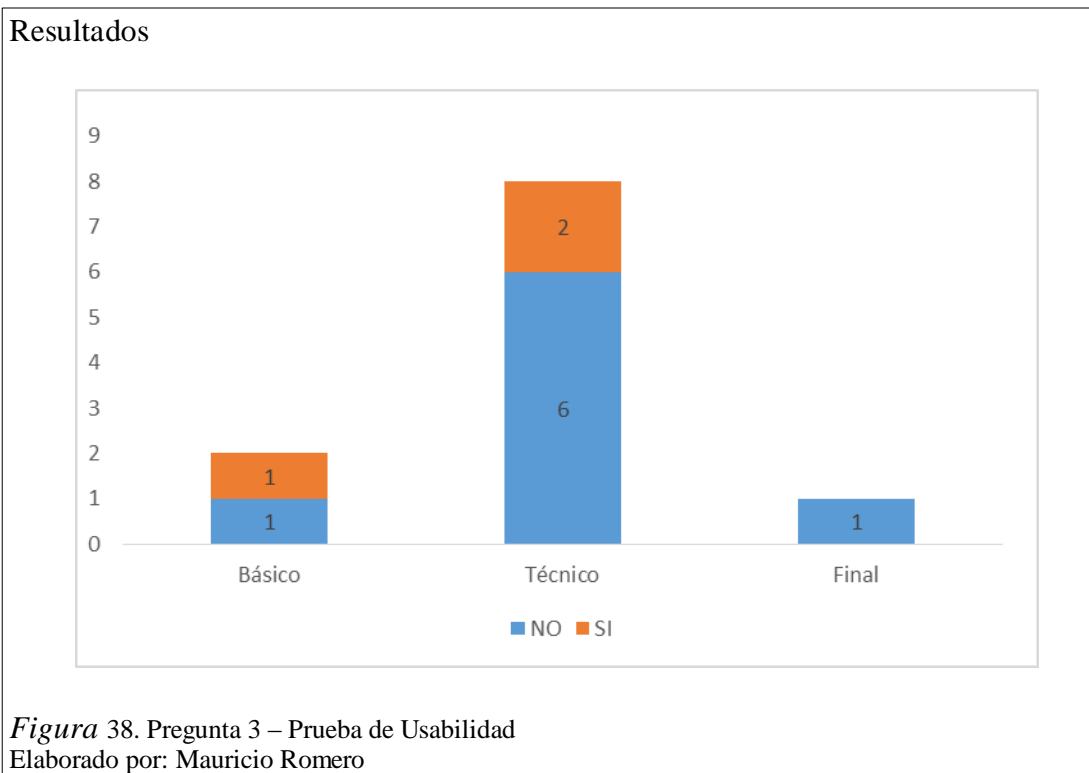
#### Resultados



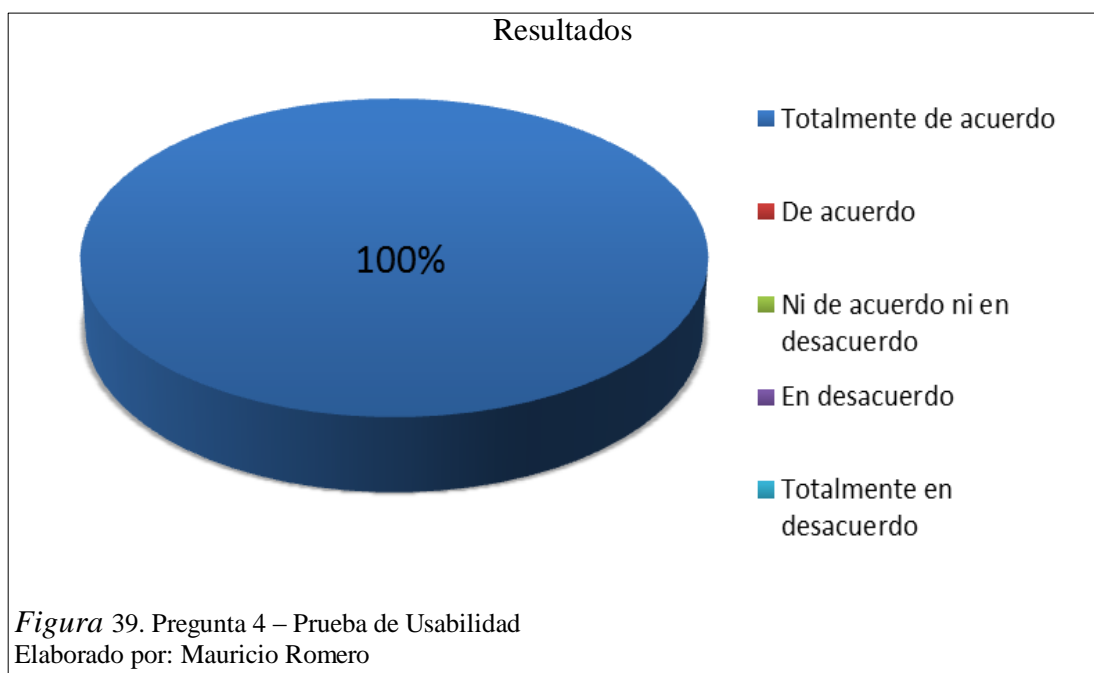
*Figura 37.* Pregunta 2 – Prueba de Usabilidad  
Elaborado por: Mauricio Romero



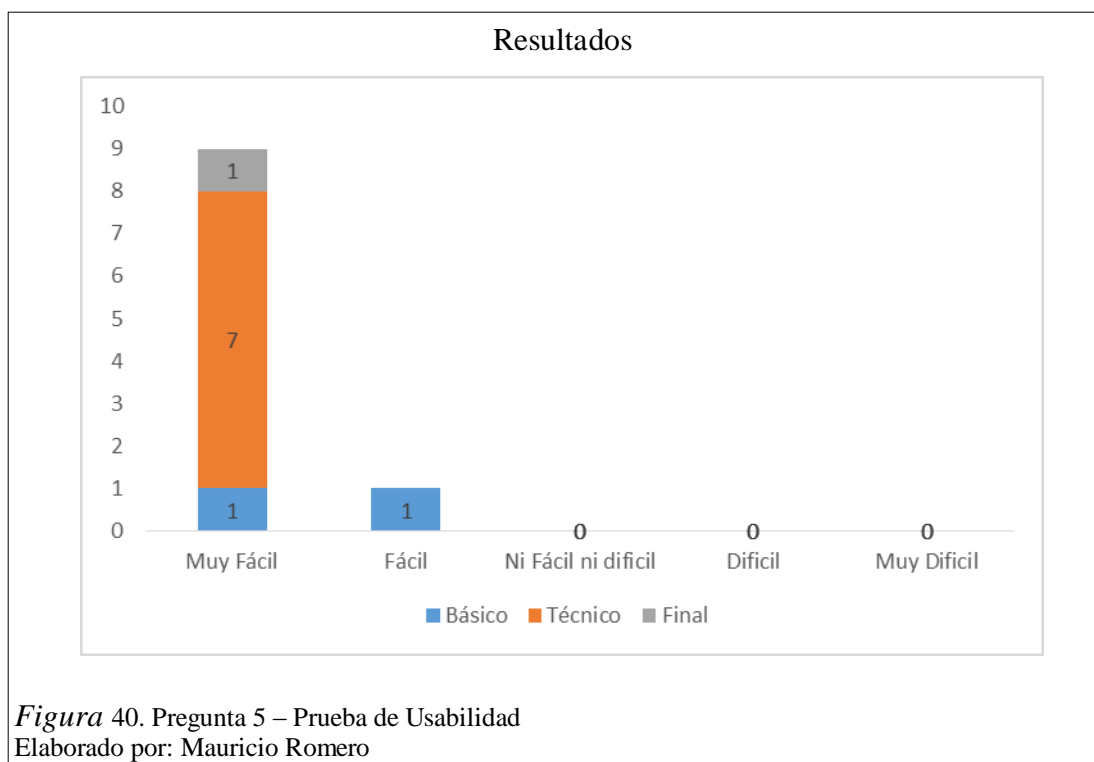
3. ¿En el módulo de Gestión de Aulas, tuvo dificultad para realizar la eliminación de asignaciones en las aulas?



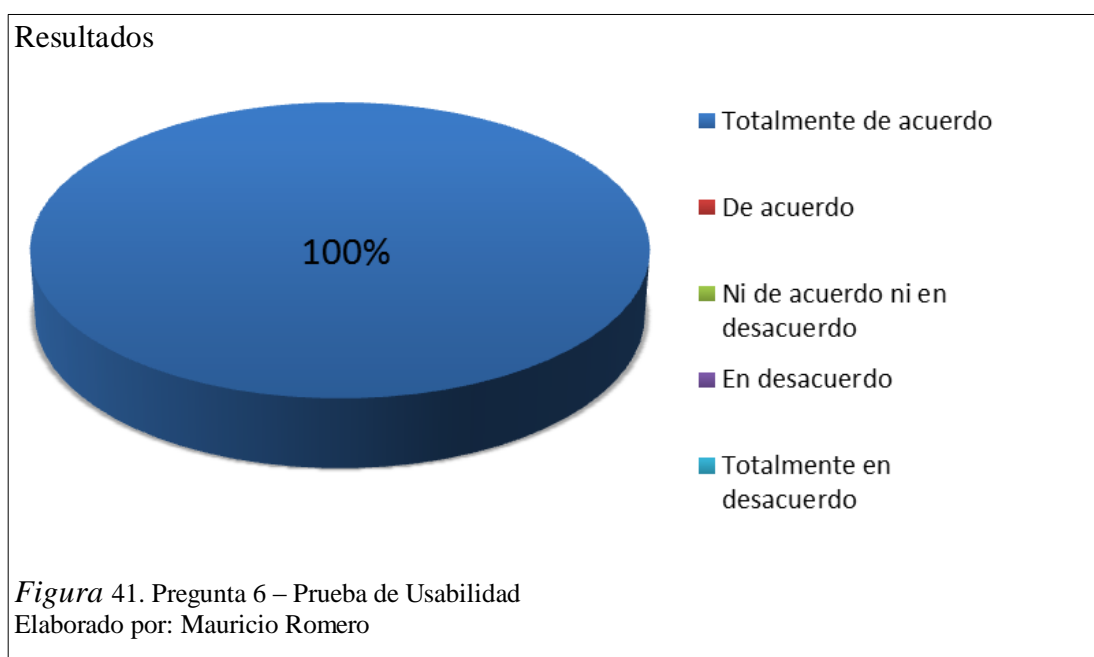
4. ¿Considera usted, que los reportes generados brindan la información necesaria?



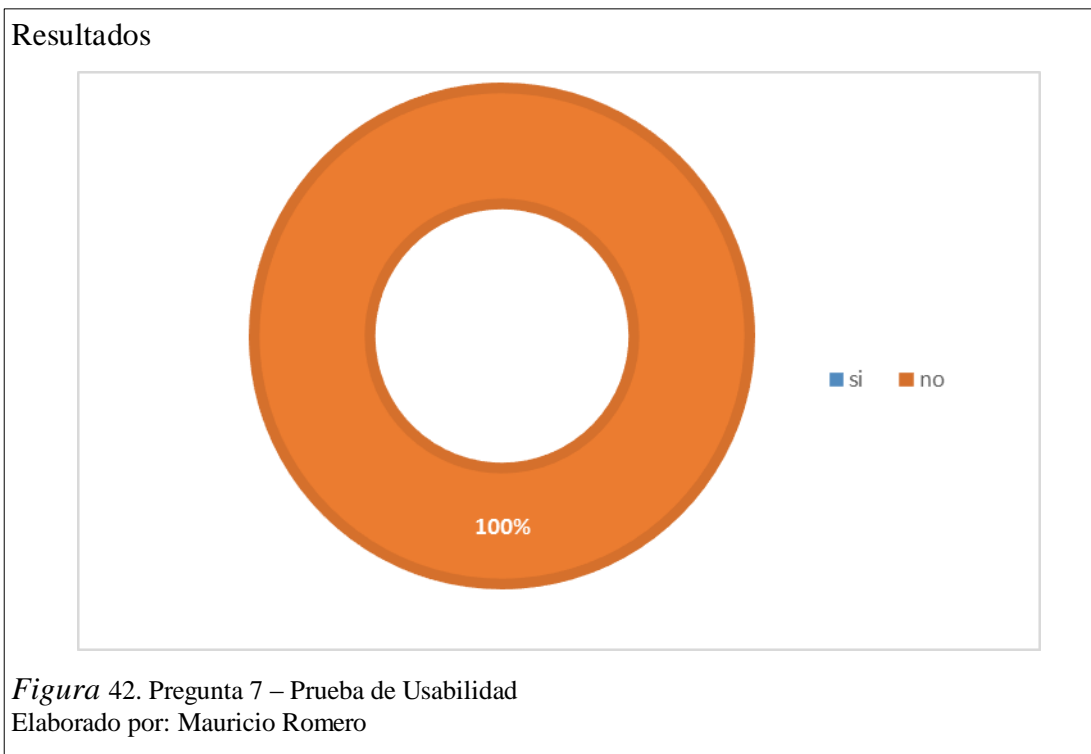
5. ¿Cuál considera usted que es el grado de facilidad de uso del Sistema de Gestión de Espacios Físicos?



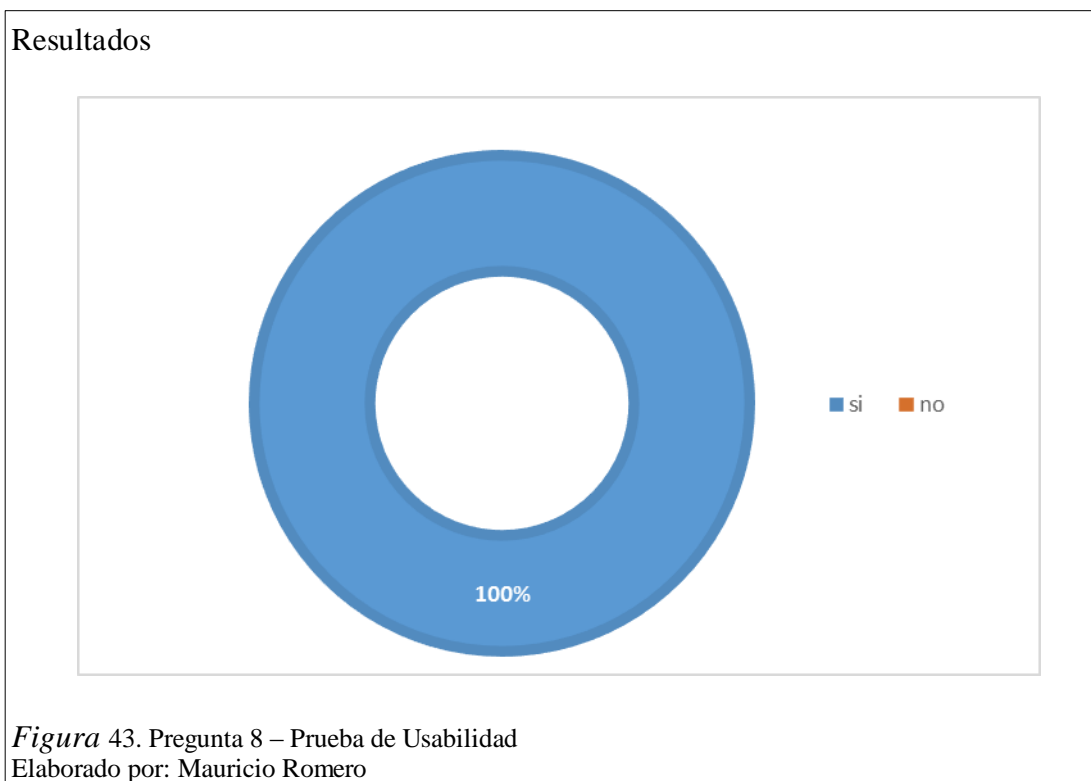
6. ¿Considera usted, que el Sistema de Gestión de Espacios Físicos cumple de forma satisfactoria las tareas requeridas?



7. ¿Durante el uso del Sistema de Gestión de Espacios Físicos, identificó usted algún error o problema que impida el cumplimiento de los objetivos para los cuáles fue diseñado el sistema?



8. ¿Considera usted, recomendar el uso de este sistema en otras sedes de la UPS?



## CONCLUSIONES

- Se obtuvo un módulo funcional que permite la asignación de aulas y espacios físicos en base a los requerimientos recabados con la Asistente Administrativa del Campus Sur, Ing. Patricia Quiroz.
- La integración con el resto de módulos del Proyecto UPS Schedule se realizó por medio de la Base de Datos, permitiendo el consumo de la información proporcionada por los otros módulos.
- Los Reportes generados por el Sistema, tuvieron un alto grado de aceptación por la información presentadas en estos, incluyendo el detalle de la asignación de espacios físicos en los horarios de las Aulas, Docentes y Grupos.
- Los procedimientos de seguridad implementados para el acceso al Sistema y la conexión a la Base de datos, garantizan la confidencialidad de la información.
- El desarrollo de interfaces amigables e intuitivas, permitieron obtener resultados positivos en la utilización del aplicativo, tal y como lo demuestran las pruebas realizadas con los usuarios.

## **RECOMENDACIONES**

- Se recomienda crear un módulo que obtenga la información directamente del Sistema SNA, para evitar el proceso de carga de información realizado en capítulo de Pruebas.
- El Proyecto UPS Schedule establece un tipo de aula por materia, se recomienda relacionar este requisito a cada hora de clase, puesto que existen materias que no utilizan un solo tipo de aula.
- Los resultados positivos obtenidos durante las pruebas realizadas en el Campus Sur, sugieren el uso de este Sistema en el resto de Campus de la Sede Quito.
- Se recomienda implementar una interfaz que permita la ejecución del Módulo de Asignación Automática bajo demanda del Asistente Administrativo por medio del Sistema de Asignación de Aulas y Espacios Físicos.

## LISTA DE REFERENCIAS

- Aguilar, L. J. (1996). Programación Orientada a Objetos. Madrid: McGRAW-HILL.
- Ambler, S. W. (2005). The Elements of UML™ 2.0 Style.
- Avila, O. (2015, 03 01). WikiHow. Retrieved from <http://es.wikihow.com/escribir-una-tesis>
- Barraza, M. O. (2006). Introducción a la programación Orientada a Objetos. Naucalpan de Juárez: Pearson Educación de México.
- Castro, L. (n.d.). Aprender Internet. Retrieved 01 15, 2015, from <http://aprenderinternet.about.com/od/Glosario/g/Que-es-Criptografia.htm>
- Centro Tecnológico CEIT-IK4. (2000, 10 16). <http://ocw.uc3m.es/ingenieria-informatica/programacion/manuales/java2-U-Navarra.pdf/view>. Retrieved from <http://ocw.uc3m.es/ingenieria-informatica/programacion/manuales/java2-U-Navarra.pdf/view>
- Ciberaula España. (n.d.). CiberAula. Retrieved 04 20, 2015, from [http://java.ciberaula.com/articulo/tecnologia\\_orientada\\_objetos/](http://java.ciberaula.com/articulo/tecnologia_orientada_objetos/)
- Cristina Alulema, C. C. (2014). Análisis, diseño, desarrollo e integración del módulo de administración para el Sistema UPS Schedule para la Universidad Politécnica Salesiana Sede Quito. Quito.
- Eckel, B. (2002). Piensa en Java. Prentice Hall.
- Garcia, I. (2003). Curso de Java. Madrid: Edi.
- Group, I. (n.d.). <http://itextpdf.com/>. Retrieved 02 01, 2015, from <http://itextpdf.com/>
- Hacienda, M. d. (n.d.). SEPP. Retrieved 04 18, 2015, from <http://www.sepg.pap.minhap.gob.es/Presup/PGE2010Ley/MaestroTomos/PGE-ROM/doc/Verificacion.pdf>
- Henry Terrero, J. P. (2010). Desarrollo de Aplicaciones con Java. Fundación de Código Libre Dominicano.

Instituto Nacional de Tecnologías de la Comunicación. (2009). INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA.

JFree. (n.d.). <http://www.jfree.org>. Retrieved 02 01, 2015, from <http://www.jfree.org/jcommon/>

Korry Douglas, S. D. (2005). PostgreSQL, 2nd Edition.

Lowagie, B. (2015). The Best iText Questions on StackOverflow.

Marco Montaguano, E. P. (2014). Análisis, diseño, desarrollo e integración del módulo de Administración de usuarios, comunicación, seguridad y auditoria del sistema UPS Schedule para la Universidad Politécnica Salesiana Sede Quito. Quito.

Martinez, R. (n.d.). [www.postgresql.org](http://www.postgresql.org). Retrieved 02 01, 2015, from [http://www.postgresql.org/es/sobre\\_postgresql](http://www.postgresql.org/es/sobre_postgresql)

Milton Chiliquinga, D. R. (2014). Análisis, diseño, desarrollo e integración del módulo de Gestión de Distributivos que trabaje de forma integrada para el Sstema UPS Schedule para la Universidad Politécnica Salesiana Sede Quito. Quito.

Paul Deitel, H. D. (n.d.). Introducción a la programación orientada a objetos.

Pérez, G. G. (2008). Aprendiendo Java y POO. Autoedición.

Proyetalis. (2015, 03 01). Proyetalis. Retrieved 05 02, 2015, from <http://www.proyetalis.com/servicios/formacion/scrum/>

Proyectos Agiles. (2013, 11 30). Proyectos Agiles. Retrieved from <http://www.proyectosagiles.org/que-es-scrum>

Silberschatz, A. (2002). FUNDAMENTOS DE BASES DE DATOS. Madrid: McGRAW-HILL.

Softeng. (2015, 01 25). Softeng. Retrieved from <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Sun Microsystem. (2015, 04 10). Retrieved from Java: [https://www.java.com/es/download/faq/whatis\\_java.xml](https://www.java.com/es/download/faq/whatis_java.xml)

Sun Microsystems. (2015, 03 01). El lenguaje de Programación Java. Retrieved from <https://docs.google.com:>

<https://docs.google.com/file/d/0Byy7aUI9u4fBRnJwc1U5VkdnaIk/edit?pli=1>

Universidad Politécnica Salesiana. (n.d.). <http://www.ups.edu.ec/>. Retrieved from <http://www.ups.edu.ec/web/guest/razon-de-ser>

Yadira Santillan, R. T. (2014). Análisis, diseño, desarrollo e integración de los módulos de Gestión Académica de Carreras y disponibilidad de Docentes, del Sistema UPS Schedule para la Universidad Politécnica Salesiana Sede Quito. Quito.



## ANEXOS

### Anexo 1. Carga Final Tbl\_usuario

Formato												
1	ID_USUARIO	ID_PERFIL	ESTADO_SESION	CEDULA	NOMBRE	NOMBRE_COMPLETO	CORREO	CORREO2	CLAVE	TEMA	FOTO	SENTENCIA
2	1	1	1	1111111111	admins	ADMINISTRADOR CAMPUS SUR	NULL	NULL	admins	aristo	men	INSERT INTO
3	2	2	1	1	JACOSTA	ACOSTA COBA JOSE MODESTO	NULL	NULL	JACOSTA	aristo	men	INSERT INTO
4	3	2	1	1	MACOSTA	ACOSTA MUNOZ MANOLO	NULL	NULL	MACOSTA	aristo	men	INSERT INTO
5	4	2	1	1	JAGUAYO	AGUAYO MORALES JOSE LUIS	NULL	NULL	JAGUAYO	aristo	men	INSERT INTO
6	5	2	1	1	AAGUILA	AGUILA TELLEZ ALEXANDER	NULL	NULL	AAGUILA	aristo	men	INSERT INTO

Figura 44. Formato carga tabla tbl\_usuario

Elaborado por: Mauricio Romero

### Anexo 2. Carga Final Tbl\_estructura

Formato														
A	B	C	D	E	F	G	H	I	J	K	L	M		
ID_ESTRUCTURA	ID_TIPO_ESTRUCTURA	ID_TIPO_AULA	ID_ASISTENTE	ID_PADRE	NOMBRE	DIRECCION	TOTAL_PISOS	NUMERO_PISOS	AREA	CAPACIDAD	OBSERVACIONES	ESTADO	SENTENCIA	
1	1	1	NULL	NULL	UPS	NULL	0	0	0	0	RAIZ	1	INSERT INTO TB	
2	2	2	NULL	NULL	1 SEDE QUITO	NULL	0	0	0	0	SEDE QUITO	1	INSERT INTO TB	
3	3	3	NULL	1	2 CAMPUS SUR	NULL	0	0	0	0	CAMPUS SUR	1	INSERT INTO TB	
4	4	4	NULL	NULL	3 BLOQUE G	NULL	3	0	0	0	BLOQUE G	1	INSERT INTO TB	
5	5	5	1	NULL	4 G1	NULL	0	1	0	0	G1	1	INSERT INTO TB	
6	5	5	1	NULL	4 G2	NULL	0	1	0	0	G2	1	INSERT INTO TB	

Figura 45. Formato carga tabla tbl\_estructura

Elaborado por: Mauricio Romero

### Anexo 3. Carga Final Tbl\_carrera

Formato

A	B	C	D	E	F	G
ID_CARRERA	ID_CAMPUS	ID_DIRECTOR	NOMBRE	COLOR	ESTADO	SENTENCIA
1	3	1	GERENCIA Y LIDERAZGO	#DEB887	1	INSERT INTO
2	3	2	INGENIERÍA AMBIENTAL	#83AF8B	1	INSERT INTO
3	3	3	INGENIERÍA CIVIL	#A7DBD8	1	INSERT INTO
4	3	4	INGENIERÍA DE SISTEMAS	#FFE4B5	1	INSERT INTO
5	3	5	INGENIERÍA ELÉCTRICA	#CCFFF	1	INSERT INTO
6	3	6	INGENIERÍA ELECTRÓNICA	#CCFFCC	1	INSERT INTO
7	3	7	INGENIERÍA MECÁNICA	#99CCFF	1	INSERT INTO

Figura 46. Formato carga tabla tbl\_carrera

Elaborado por: Mauricio Romero

### Anexo 4. Carga Final Tbl\_materia

Formato

A	B	C	D	E	F	G	H	I	J	K	
ID_MATERIA	ID_CARRERA	ID_PARACADEMICO	ID_NIVEL	ID_MALLA	ID_TIPO_AULA	CODIGO	NOMBRE	CREDITOS	ESTADO	ID_MENCION	SENTENCIA
1	1	NULL	2	1	1	9426	ALGEBRA LINEAL	4	1	0	INSERT INTO T
2	1	NULL	2	1	1	9428	BASES DE LA NEGOCIACION	2	1	0	INSERT INTO T
3	1	NULL	2	1	2	9430	CONTABILIDAD GENERAL	3	1	0	INSERT INTO T
4	1	NULL	2	1	1	9442	ESPIRITUALIDAD JUVENIL SALESIANA	2	1	0	INSERT INTO T

Figura 47. Formato carga tabla tbl\_materia

Elaborado por: Mauricio Romero

## Anexo 5. Carga Final Tbl\_distributivo

Formato

A	B	C	D	E	F	G	
ID_DISTRIBUTIVO	ID_CARRERA	ID_USUARIO	ID_MATERIA	ID_GRUPO	ID_PERIODO	ESTADO	
1	1	2	25	2	2	1	INSERT
2	1	2	33	2	2	1	INSERT
3	1	2	26	2	2	1	INSERT
4	1	2	17	2	2	1	INSERT
5	1	3	11	2	2	1	INSERT
6	1	6	55	2	2	1	INSERT
7	1	6	37	2	2	1	INSERT

Figura 48. Formato carga tabla tbl\_distributivo

Elaborado por: Mauricio Romero

## Anexo 6. Carga Final Tbl\_horario

Formato

A	B	C	D	E	F	G	H	I	
ID_HORARIO	ID_CARRERA	ID_ESTADOH	ID_DISTRIBUTIVO	ID_AULA	ID_HORA	ID_DIA	OBSERVACION		
1	1	1	1	NULL	3	6		INSERT INTO TBL	
2	1	1	1	NULL	4	6		INSERT INTO TBL	
3	1	1	2	NULL	11	2		INSERT INTO TBL	
4	1	1	2	NULL	1	6		INSERT INTO TBL	


Figura 49. Formato carga tabla tbl\_horario

Elaborado por: Mauricio Romero

## Anexo 7. Reporte por aula

### Reporte

# BLOQUE A



**UNIVERSIDAD POLITÉCNICA**  
**SALESIANA**  
ECUADOR

P46

A1

HORA	LUNES			MARTES			MIÉRCOLES			JUEVES			VIERNES			SABADO		
07:00	IS	II G: 2	CALCULO INTEGRAL	IS	I G: 3	TECNICAS DE INVESTIGACION	IS	II G: 3	ANTROPOLOGIA CRISTIANA	IS	I G: 1	CALCULO DIFERENCIAL	IS	XI G: 1	DEONTOLOGIA	IS	XI G: 1	INGENIERIA DEL TRAFICO (T)
08:00	IS	II G: 2	CALCULO INTEGRAL	IS	I G: 3	TECNICAS DE INVESTIGACION	IS	II G: 3	ANTROPOLOGIA CRISTIANA	IS	I G: 1	CALCULO DIFERENCIAL	IS	XI G: 1	DEONTOLOGIA	IS	XI G: 1	INGENIERIA DEL TRAFICO (T)
09:00	IS	I G: 1	ALGEBRA LINEAL	IS	III G: 1	ESPIRITUALIDAD JUVENIL SALESIANA	IS	I G: 1	CALCULO DIFERENCIAL	IS	VI G: 2	ETICA	IS	I G: 3	CALCULO DIFERENCIAL	IS	X G: 1	DISEÑO DE REDES (T)
10:00	IS	I G: 1	ALGEBRA LINEAL	IS	III G: 1	ESPIRITUALIDAD JUVENIL SALESIANA	IS	I G: 1	CALCULO DIFERENCIAL	IS	VI G: 2	ETICA	IS	I G: 3	CALCULO DIFERENCIAL	IS	X G: 1	DISEÑO DE REDES (T)
11:00	IS	II G: 1	CALCULO INTEGRAL	IS	IV G: 1	MATEMATICAS SUPERIORES	IS	I G: 3	CALCULO DIFERENCIAL	IS	I G: 1	ALGEBRA LINEAL	IS	I G: 1	CALCULO DIFERENCIAL			
12:00	IS	II G: 1	CALCULO INTEGRAL	IS	IV G: 1	MATEMATICAS SUPERIORES	IS	I G: 3	CALCULO DIFERENCIAL	IS	I G: 1	ALGEBRA LINEAL	IS	I G: 1	CALCULO DIFERENCIAL			
13:30																		
14:30																		
15:30																		
16:30																		
17:30	IS	VIII G: 1	CONTROL DE CALIDAD (O)	IS	XI G: 1	INVESTIGACION DE OPERACIONES	IS	VIII G: 1	ADMINISTRACION Y GESTION DE REDES (T)	IS	X G: 1	DISEÑO DE REDES (T)	IS	X G: 1	SISTEMAS OPERACIONALES (O)			
18:30	IS	VIII G: 1	CONTROL DE CALIDAD (O)	IS	XI G: 1	INVESTIGACION DE OPERACIONES	IS	VIII G: 1	ADMINISTRACION Y GESTION DE REDES (T)	IS	X G: 1	DISEÑO DE REDES (T)	IS	X G: 1	SISTEMAS OPERACIONALES (O)			
19:30				IS	XI G: 1	SISTEMAS EXPERTOS (O)	IS	XI G: 1	INVESTIGACION DE OPERACIONES	IS	XI G: 1	SISTEMAS EXPERTOS (O)	IS	VIII G: 1	ADMINISTRACION Y GESTION DE REDES (T)			
20:30				IS	XI G: 1	SISTEMAS EXPERTOS (O)	IS	XI G: 1	INVESTIGACION DE OPERACIONES	IS	XI G: 1	SISTEMAS EXPERTOS (O)	IS	VIII G: 1	ADMINISTRACION Y GESTION DE REDES (T)			

IS

INGENIERÍA DE SISTEMAS

IS INGENIERÍA DE SISTEMAS

Figura 50. Reporte por aula  
Elaborado por: Mauricio Romero

Anexo 8. Reporte por docente

Reporte

VILLAGOMEZ MENENDEZ JOSE LUIS



UNIVERSIDAD POLITÉCNICA  
SALESIANA  
ECUADOR

P46

HORA	LUNES			MARTES			MIÉRCOLES			JUEVES			VIERNES	SABADO
07:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
08:00	L5	II 1	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 2	PROGRAMACION II	L2	II 2	PROGRAMACION II		
09:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
10:00	L4	IV 1	SISTEMAS OPERATIVOS I	L3	II 2	PROGRAMACION II	L2	IV 1	SISTEMAS OPERATIVOS I	L2	II 1	PROGRAMACION II		
11:00				L2	II 1	PROGRAMACION II								
12:00				L2	II 1	PROGRAMACION II								
13:30														
14:30														
15:30														
16:30														
17:30														
18:30														
19:30														
20:30														

IS


INGENIERÍA DE SISTEMAS

Figura 51. Reporte por docente  
Elaborado por: Mauricio Romero

Anexo 9. Reporte por Grupo

Reporte

INGENIERÍA DE SISTEMAS

UNIVERSIDAD POLITÉCNICA  
SALESIANA  
ECUADOR

P46

Nº SEPTIMO

Grupo: TELEMÁTICA

HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SABADO
07:00						
08:00						
09:00						<div>BLOQUE A</div> <div>P: 4L2</div> TCP/IP
10:00						<div>BLOQUE A</div> <div>P: 4L2</div> TCP/IP
11:00						<div>BLOQUE A</div> <div>P: 4L1</div> REDES II
12:00						<div>BLOQUE A</div> <div>P: 4L1</div> REDES II
13:30						
14:30						
15:30						
16:30						
17:30	<div>BLOQUE A</div> <div>P: 1A2</div> TEORIA ELECTROMAGNETICA (T)	<div>BLOQUE A</div> <div>P: 4L4</div> PROGRAMACION HYPERMEDIAL	<div>BLOQUE A</div> <div>P: 1A2</div> TEORIA ELECTROMAGNETICA (T)	<div>BLOQUE A</div> <div>P: 4L7</div> SISTEMAS DE COMUNICACION I (T)	<div>BLOQUE A</div> <div>P: 4L6</div> PROGRAMACION HYPERMEDIAL	
18:30	<div>BLOQUE A</div> <div>P: 1A2</div> TEORIA ELECTROMAGNETICA (T)	<div>BLOQUE A</div> <div>P: 4L4</div> PROGRAMACION HYPERMEDIAL	<div>BLOQUE A</div> <div>P: 1A2</div> TEORIA ELECTROMAGNETICA (T)	<div>BLOQUE A</div> <div>P: 4L7</div> SISTEMAS DE COMUNICACION I (T)	<div>BLOQUE A</div> <div>P: 4L6</div> PROGRAMACION HYPERMEDIAL	
19:30	<div>BLOQUE A</div> <div>P: 4L3</div> REDES II	<div>BLOQUE A</div> <div>P: 4L5</div> TCP/IP	<div>BLOQUE A</div> <div>P: 4L3</div> SISTEMAS DE COMUNICACION I (T)		<div>BLOQUE A</div> <div>P: 4L1</div> TCP/IP	
20:30	<div>BLOQUE A</div> <div>P: 4L3</div> REDES II	<div>BLOQUE A</div> <div>P: 4L5</div> TCP/IP	<div>BLOQUE A</div> <div>P: 4L3</div> SISTEMAS DE COMUNICACION I (T)		<div>BLOQUE A</div> <div>P: 4L1</div> TCP/IP	

Figura 52. Reporte por grupo  
Elaborado por: Mauricio Romero